

MathScript

Simon Ihmig

COLLABORATORS

	<i>TITLE :</i> MathScript		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Simon Ihmig	March 28, 2025	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	MathScript	1
1.1	MathScript Hilfe	1
1.2	Beschreibung	1
1.3	system-anforderungen@{par}	2
1.4	Starten des Programms	2
1.5	Eingabe der Formeln	2
1.6	Objekte	3
1.7	Bedienung von MathScript	3
1.8	Auswahl-Gadgets	3
1.9	Verknüpfungen	3
1.10	Pfeile	4
1.11	Logik-Symbole	4
1.12	Klammern	4
1.13	Summe,Produkt und Integral	4
1.14	kleine griechische Buchstaben	4
1.15	große griechische Buchstaben	4
1.16	Vergleichsoperatoren	4
1.17	Symbole der Mengenlehre	4
1.18	Symbole	4
1.19	Hoch-/Tiefstellen	4
1.20	Brüche und Wurzeln	5
1.21	Akzente	5
1.22	Eingabefeld	5
1.23	menus	5
1.24	Neu	5
1.25	Laden ...	5
1.26	Speichern	5
1.27	Speichern als ...	5
1.28	Exportieren	6
1.29	eps	6

1.30	info	6
1.31	Beenden	6
1.32	ARexx-Script starten ...	6
1.33	Einstellungen ändern	6
1.34	Einstellungen speichern	7
1.35	arexx	7
1.36	clear	7
1.37	getcontents	7
1.38	setcontents	8
1.39	getposition	8
1.40	setposition	8
1.41	insert	8
1.42	gotonextobject	8
1.43	gotoprevobject	8
1.44	Load	8
1.45	Save	8
1.46	Export	9
1.47	Info	9
1.48	version	9
1.49	screentofront	9
1.50	requester	9
1.51	Quit	9
1.52	getprefs	9
1.53	setprefs	9
1.54	codes	10
1.55	Nutzungsbedingungen	10
1.56	registrierung	11
1.57	Registrierungs-Formular	11
1.58	zukunft	11

Chapter 1

MathScript

1.1 MathScript Hilfe

MathScript © Copyright Simon Ihmig 1994 MathScript ist SHAREWARE !

Anleitung zu MathScript

1. Einführung

1.1 **Beschreibung** 1.2 System-Anforderungen

2. Benutzung des Programms

2.1 **Starten des Programms** 2.2 **Eingabe der Formel** 2.3 **Bedienung des Programms**

3. Anhang

3.1 **ARexx-Schnittstelle** 3.2 **Steuer-Codes**

Nutzungsbedingungen **Registrierung** **Die Zukunft**

Meine Adresse:

Simon Ihmig Beim Rauhen Hause 30 22111 Hamburg Germany e-mail: Internet: Ihmig@tu-harburg.d400.de Z-Netz: GrafZahl@Online

1.2 Beschreibung

MathScript ist ein Formel-Editor, der dazu dient, mathematische Formeln in eine Textverarbeitung zu integrieren. Das ist an sich nichts Neues, stellt jedoch auf dem Amiga eine Marktlücke dar. Sich wegen der mathematischen Formeln in TeX einzuarbeiten, wird nicht jedermanns Sache sein.

Deswegen entstand MathScript.

Mit diesem Programm ist es auf leichte Weise möglich, komplexe Formeln zu generieren. Diese werden als Grafik-Datei abgespeichert, um sie in Textverarbeitungs- und DTP-Programmen verwenden zu können.

Um höchste Qualität zu garantieren, werden die Formeln als Vektor-Grafik im EPS-Format abgespeichert. Diese kann dann in allen Textverarbeitungs- und DTP-Programme, die dieses Format unterstützen (FinalWriter, PageStream...) weiterverwendet werden. Sie können außerdem auf PostScript-fähigen Druckern und durch softwaremäßige PostScript-Interpreter auf jedem Drucker ausgedruckt werden.

In der Version 1.0 bietet MathScript folgende Funktionen:

- umfangreiches Sortiment an mathematischen und physikalischen Symbolen
 - Steuerzeichen zur Formatierung (hoch-, tiefstellen, Brüche, Wurzeln...)
 - Komfortable Eingabe über Workbench GUI
-

- Exportieren im EPS-Format (Encapsuled PostScript)
- ARexx-Port
- Unterstützung der locale.library (ab OS 2.1). Verfügbare Sprachen: deutsch, englisch, französisch. Sie können auch selbst neue Sprachen hinzufügen (siehe Catalogs/readme).

1.3 system-anforderungen@{par}

MathScript benötigt zum Laufen mindestens OS 2.0. Ab OS 2.1 unterstützt es lokalisierte Benutzerführungen.

Um den ARexx-Port von MathScript zu nutzen, muß der ARexx-Interpreter REXXMAST laufen.

1.4 Starten des Programms

Starten von MathScript ...

von der Workbench:

Um MathScript von der Workbench aus zu starten, gehen Sie in das Verzeichnis, wo sich das Programm befindet, und klicken sie das zugehörige Piktogramm mit der Maus an. Alternativ können Sie auch ein Piktogramm einer abgespeicherten Formel anklicken, wobei dann diese Formel automatisch geladen wird.

von der Shell:

Um MathScript von der Shell zu starten, geben Sie "MathScript:MathScript" ein.

Dabei werden folgende Argumente unterstützt:

FILE,PUBSCREEN/K

FILE : Name einer zu ladenden Formel

PUBSCREEN : Name eines öffentlichen Bildschirms, auf dem MathScript geöffnet werden soll

1.5 Eingabe der Formeln

Die Eingabe der Formeln erfolgt ausschließlich über die Eingabe in das **Eingabefeld**. Diese kann über die Benutzung der Tastatur oder der **Auswahl-Gadgets** erfolgen. Dabei werden zur Formatierung der Symbole und Buchstaben spezielle **Steuer-Codes** benutzt, also ohne WYSIWYG-Darstellung.

Um Zeichen des Standard-ASCII-Zeichensatzes (alle Zeichen, die auf der Tastatur abgebildet sind, aber ohne die dt. Umlaute!) in die Formel einzufügen, können diese direkt mit der Tastatur bei aktivem Eingabefeld (zum Aktivieren ins Fenster klicken) eingegeben werden.

Die **Steuer-Codes** können wahlweise über die Tastatur oder die **Auswahl-Gadgets** eingefügt werden.

Mathematische Sonderzeichen müssen über die Auswahl-Gadgets ausgewählt werden.

Grundsätzliches zur Benutzung der **Steuer-Codes**:

Steuer-Codes bewirken die Formatierung der ihnen zugehörigen **Objekte**. Das kann z.B. das Hochstellen von Zeichen zum Potenzieren (x^2) oder die Bildung eines Bruches sein. Sie sind in der fertig als Grafik abgespeicherten Formel nicht mehr zu sehen.

Nehmen wir uns ein Beispiel vor:

Um einen Bruch darzustellen, gibt es das Steuerzeichen '/'. Seine Syntax lautet:

Bruch: Syntax: <Obj1>/<Obj2> Funktion: Erzeugt einen Bruch. Eingabe: <Obj1> - Zähler <Obj2> - Nenner

Die Eingabe von 1/2 bewirkt z.B. folgende Darstellung in der fertigen Grafikdatei:

1 - 2

Wie einzelne Zeichen als Objekt zusammengefaßt werden, erfahren Sie in **Objekte**.

Eine komplette Übersicht der unterstützten Formatierungsoptionen mitsamt ihrer Syntax finden Sie unter **Steuer-Codes**.

Ist die Formel ihren Wünschen entsprechend eingegeben, können Sie in den **Menüs** die Formel als Textdatei abspeichern, um sie später weiterverwenden zu können, oder gleich als Grafikdatei exportieren, wo sie ihr eigentliches Aussehen erhält.

1.6 Objekte

Die **Steuer-Codes** benötigen Objekte zur Formatierung der Formel. So ein Objekt ist im einfachsten Fall ein einzelner Buchstabe.

So erzeugt die Eingabe von $1/2x$ folgenden Term in der fertigen Grafik: $1 - x^2$. Möchte man nun aber, daß der Nenner $2x$ ist, also mehrere Zeichen verwenden, so muß man diese zu einem Objekt zusammenfassen. Dies geschieht, indem man diejenigen Zeichen, die als ein Objekt gelten sollen, innerhalb eckiger Klammern schreibt. Also erzeugt die Eingabe von $1/[2x]$ diesen Term: $1 -- 2x$. Objekte lassen sich (fast) beliebig oft verschachteln. Um einen Doppelbruch zu erstellen, gibt man z.B. $[1/x]/3$ ein, was zu folgender Ausgabe führt: $1 - x --- 3$

1.7 Bedienung von MathScript

MathScript öffnet immer das Hauptfenster. In diesem erfolgt die **Eingabe** der Formeln.

Dazu werden die **Auswahl-Gadgets**, eine Gruppe von 13 Gadgets, und unterhalb dieser das **Eingabefeld** benutzt.

Die sonstigen Funktionen von MathScript sind über die **Menüs** zugänglich.

Für erfahrene Benutzer steht außerdem eine **ARexx-Schnittstelle** zur Verfügung.

1.8 Auswahl-Gadgets

Die Auswahl-Gadgets bestehen aus 13 Gadgets. Jedes von diesen öffnet ein Auswahl-Fenster. In diesen Auswahl-Fenstern befinden sich zahlreiche weitere Gadgets, die mit verschiedenen Symbolen belegt sind. Klickt man auf eines dieser Symbole, wird es an die aktuelle Position im **Eingabefeld** eingefügt. Danach wird das Fenster wieder geschlossen.

Es stehen folgende Gadgets zur Verfügung:

Verknüpfungen Pfeile Logik Klammern Sum,Prod,Int griech. Zeichen GRIECH. ZEICHEN

Vergleichsop. Mengensymbole Symbole Hoch-/Tiefstellen Bruch,Wurzel Akzente

Sie werden durch eine Auswahl ihrer Unter-Gadgets repräsentiert.

Neben einfachen Zeichen können auch **Steuer-Codes** eingefügt werden. Diese werden in einer symbolischen Darstellung angezeigt (also nicht in ihrer Schreibweise). Dabei werden ihre zugehörigen **Objekte** folgendermaßen symbolisiert: steht für ein zusammengesetztes Objekt, # steht für ein einzelnes Zeichen.

Beispiel: Das Symbol # steht für Hochstellen. Es fügt folgende Zeichenkette in das **Eingabefeld** ein: $\hat{[]}$. Nehmen wir an, es steht dort bereits ein x . Nach dem Klicken auf das Symbol steht dort $x^{\hat{[]}}$. Der Cursor wird automatisch hinter das erste zusammengesetzte **Objekt** gesetzt, also hinter '['. Durch Eingabe einer '2' lautet der Inhalt " $x^{\hat{[2]}}$ ", was als Grafik zu dieser Darstellung führt: x^2 . Wenn das erste Objekt jedoch ein zusammengesetztes ist, so muß dessen Inhalt nach dem Anwählen des Gadgets eingegeben werden, damit es von den eckigen Klammern umgeben ist (siehe **Objekte**).

1.9 Verknüpfungen

Enthält eine Sammlung mathematischer Verknüpfungen und Vorzeichen.

1.10 Pfeile

Beinhaltet je 5 einfache und doppelte Pfeile.

1.11 Logik-Symbole

Enthält prädikaten- und aussagenlogische Symbole.

1.12 Klammern

Beinhaltet **Steuer-Codes** zur Erzeugung von Klammern variabler Größe. Von jeder Klammer-Art ($\{, (, [,], \rangle$) gibt es sowohl beidseitige, linksseitige als auch rechtsseitige Versionen.

1.13 Summe, Produkt und Integral

Je ein Zeichen zur Bildung einer Summe, eines Produkts und eines Integrals.

1.14 kleine griechische Buchstaben

Enthält sämtliche kleinen griechischen Buchstaben.

1.15 große griechische Buchstaben

Enthält sämtliche großen griechischen Buchstaben.

1.16 Vergleichsoperatoren

Beinhaltet die wichtigsten Vergleichsoperatoren.

1.17 Symbole der Mengenlehre

Eine Sammlung der Zeichen aus der Mengenlehre.

1.18 Symbole

Enthält sonstige Symbole, die keiner anderen Gruppe angehören.

1.19 Hoch-/Tiefstellen

Faßt alle **Steuer-Codes** zum Positionieren von **Objekten** zusammen.

1.20 Brüche und Wurzeln

Enthält **Steuer-Codes** zur Bruch- und Wurzel-Bildung.

1.21 Akzente

Enthält **Steuer-Codes** zum Bilden von mathematisch/physikalischen Akzenten wie Ableitungen nach x und t und Vektor-Pfeile.

1.22 Eingabefeld

In das Eingabefeld erfolgt die Eingabe der Formel. Es ist ein String-Gadget und akzeptiert daher alle Tastenkombinationen, die auch auf andere String-Gadgets angewendet werden können. Zusätzlich kann man mit der TAB-Taste zum nächsten, mit der SHIFT-TAB Kombination zum vorherigen zusammengesetzten **Objekt** springen.

1.23 menus

Übersicht aller Menupunkte:

Projekt ARexx Einstellungen **Neu ARexx-Script starten...** **Einstellungen ändern...** **Laden...** **Einstellungen speichern** **Speichern**
Speichern als... **Exportieren EPS** **Über...** **Beenden**

1.24 Neu

Löscht das **Eingabefeld**.

Achtung: Nach Anwahl dieses Menupunktes ist der Inhalt des Eingabefeldes unwiederbringlich verloren !

1.25 Laden ...

Öffnet einen Datei-Requester, in dem eine Datei ausgewählt werden kann, die in das **Eingabefeld** geladen wird. Dabei werden nur Dateien angezeigt, die die Endung .fml haben. Diese wird automatisch beim **Speichern** angehängt. Wollen Sie eine ASCII-Datei einladen, die diese Endung nicht hat, so löschen Sie das "Muster"-Gadget.

1.26 Speichern

Führt dieselbe Operation wie **Speichern als ...** aus, mit dem Unterschied, daß, falls eine Datei schon geladen oder gespeichert wurde, kein Datei-Requester erscheint, sondern die aktuelle Datei zum Speichern gewählt wird.

Der Dateiname der aktuellen Datei wird in der Titelleiste des Fensters angezeigt !

1.27 Speichern als ...

Dieser Menupunkt dient zum Speichern des Inhalts des **Eingabefeldes** als Formel-Datei, damit diese Formel später wiederverwendet bzw. verändert werden kann. Es wird ein Datei-Requester geöffnet, in dem eine Datei ausgewählt werden kann, in die die Formel geschrieben wird.

Alle mit MathScript abgespeicherten Formeln erhalten die Endung .fml, um sie von anderen Dateien zu unterscheiden.

1.28 Exportieren

Dieser Ober-Menupunkt beinhaltet den eigentlichen Hauptteil des Programms, nämlich die Umformung der eingegebenen Formel in eine Grafik.

Als Grafik-Format steht in dieser Version von MathScript folgendes zur Verfügung: **EPS**-Format

1.29 eps

Dieser Unter-Menupunkt speichert die Formel als Vektor-Grafik im EPS-Format ab. Er öffnet ein Datei-Requester, in dem der Dateiname ausgewählt werden kann, unter dem die Formel abgespeichert wird.

Dieses Format ist das am meisten verbreitete Vektor-Grafik-Format. Im Gegensatz zu BitMap-Formaten wie das IFF-ILBM-Format zeichnen sich diese Vektor-Grafiken durch tadellose Druckqualität und besonders durch Beibehalten der Qualität beim Skalieren (Vergrößern) aus.

EPS bedeutet Encapsuled PostScript, was so viel heißt wie eingebettetes PostScript. PostScript ist eine interpretative Seitenbeschreibungssprache, die von einigen Laser-Druckern und von Belichtungs-Maschinen verstanden wird. Wenn Sie kein solches PostScript-fähiges Ausgabegerät haben (ich hab's auch nicht), sind Sie auf die Software angewiesen, mit der Sie die Formeln weiterverwenden wollen. Das Textverarbeitungsprogramm "FinalWriter" z.B. kann diese Vektor-Grafik rastern, um Sie dann auf gewöhnlichen Druckern auszudrucken.

Alle in MathScript abgespeicherten EPS-Dateien erhalten die Endung .ps.

1.30 info

Öffnet ein Informationsfenster, in dem Angaben zum Programm und zu mir stehen.

1.31 Beenden

Beendet das Programm nach einer eventuellen Sicherheitsabfrage.

1.32 ARexx-Script starten ...

Es wird ein Datei-Requester geöffnet, in dem ein ARexx-Script ausgewählt werden kann, das ausgeführt werden soll.

Genauere Information zur ARexx-Schnittstelle von MathScript gibt es unter **ARexx**.

1.33 Einstellungen ändern

Dieser Menupunkt öffnet ein Fenster, in dem MathScript konfiguriert werden kann. Es stehen folgende Optionen zur Verfügung:

Bildschirm Mit dem Cycle-Gadget kann man zwischen den verschiedenen Bildschirm-Typen wählen, auf denen die Benutzeroberfläche von MathScript laufen soll. Workbench : Öffnet die MathScript-Fenster auf der Workbench. Standard öffentlicher Bildschirm : Benutzt den standardmäßigen öffentlichen Bildschirm. Dies ist in der Regel die Workbench. öffentlicher Bildschirm : Benutzt den öffentlichen Bildschirm, dessen Name in dem Gadget rechts daneben eingegeben ist. Wenn beim Start vom CLI ein öffentlicher Bildschirm angegeben wird, wird diese Einstellung nicht beachtet.

Pfad Mit dem kleinen Datei-Requester-Gadget oder dem Eingabe-Gadget links daneben kann ein Pfad angegeben werden, in dem MathScript normalerweise seine Formeln lädt und speichert.

PS-Font einbetten Ist diese Option aktiviert, wird die PostScript-Beschreibung des mathematischen Zeichensatzes in die EPS Datei eingebettet. Dies ist notwendig, wenn der PostScript-Interpreter, den Sie benutzen, Fonts nicht von Ihrer Festplatte nachladen kann. Dies ist selbstverständlich bei allen PostScript-fähigen Druckern der Fall. Wird dagegen eine Anwendung mit software-seitigem Interpreter (Post, PostView, FinalWriter...) benutzt, sucht diese (bei entsprechender init.ps Datei) im Verzeichnis PSFont: nach verfügbaren Zeichensätzen. Daher empfiehlt es sich in diesem Fall, den Font nicht mit einzubinden, um die Datei nicht unnötig zu vergrößern.

PS-Font Höhe In diesem Gadget kann man die Höhe des PostScript-Fonts in pt bestimmen.

Icons erstellen Ist diese Option aktiviert, erstellt MathScript beim Speichern und Exportieren der Formeln passende Piktogramme.

Um die gemachten Einstellungen zu akzeptieren, klicken Sie auf "Ok", bei "Abbrechen" werden sie wieder rückgängig gemacht.

1.34 Einstellungen speichern

Speichert die aktuellen Einstellungen ab. Diese werden dann beim nächsten Start automatisch benutzt.

1.35 arexx

Über den ARexx-Port von MathScript kann das Programm von außen, d.h. ohne Einwirkung des Benutzers, gesteuert werden. Damit sind Makros und Scripts, die z.B. eine Vorschau-Anzeige liefern oder die Formeln automatisch in eine Textverarbeitung einbinden, kein Problem. Um MathScript über den ARexx-Port zu steuern, muß der REXXMaster laufen (siehe System-Dokumentation). Der Name des ARexx-Ports lautet: "MathScript".

Ein ARexx-Script kann von MathScript aus durch den Menüpunkt **ARexx-Script starten ...** gestartet werden.

Argumente in spitzen Klammern < ... > müssen angegeben werden, Argumente in eckigen Klammern [...] können angegeben werden.

Hier nun die Liste aller unterstützten Kommandos:

Clear GetContents

SetContents GetPosition

SetPosition Insert

GotonextObject GotoprevObject

Load Save

Export Info

Version ScreenToFront

LoadRequester Quit

GetPrefs SetPrefs

1.36 clear

Clear: Syntax: Clear Funktion: Löscht das **Eingabefeld**. Entspricht dem Menüpunkt **Neu**. Eingabe: - Ergebnis: -

1.37 getcontents

GetContents: Syntax: GetContents Funktion: Liefert den aktuellen Inhalt des **Eingabefeldes**. Eingabe: - Ergebnis: result - Inhalt

1.38 setcontents

SetContents: Syntax: SetContents <Inhalt> Funktion: Belegt den Inhalt des **Eingabefeldes**. Eingabe: <Inhalt> - neuer Inhalt des Eingabefeldes Ergebnis: -

1.39 getposition

GetPosition: Syntax: GetPosition Funktion: Liefert die Position des Cursors im **Eingabefeld** Eingabe: - Ergebnis: result - Position des Cursors

1.40 setposition

SetPosition: Syntax: SetPosition <Position> Funktion: Setzt den Cursor auf eine Position. Eingabe: <Position> - neue Position des Cursors Ergebnis: -

1.41 insert

Insert: Syntax: Insert <String> Funktion: Fügt eine Zeichenkette hinter die aktuelle Cursor-Position ins Eingabefeld ein. Eingabe: <String> - Zeichenkette, die eingefügt werden soll Ergebnis: -

1.42 gotonextobject

GotonextObject: Syntax: GotonextObject Funktion: Setzt den Cursor in das nächste zusammengesetzte **Objekt**. Entspricht dem Drücken der TAB-Taste. Eingabe: - Ergebnis: -

1.43 gotoprevobject

GotoprevObject: Syntax: GotoprevObject Funktion: Setzt den Cursor in das vorherige zusammengesetzte **Objekt**. Entspricht dem Drücken der SHIFT-TAB-Tasten. Eingabe: - Ergebnis: -

1.44 Load

Load: Syntax: Load [Dateiname] [FORCE] Funktion: Lädt eine Text-Datei als Formel. Wird [Dateiname] nicht angegeben, wird ein Datei-Requester geöffnet. Eingabe: [Dateiname] - Name der zu ladenden Datei [FORCE] - verhindert die Ausgabe einer Warnung Ergebnis: -

1.45 Save

Save: Syntax: Save [Dateiname] [FORCE] Funktion: Speichert die aktuelle Formel als Text-Datei. Wird [Dateiname] nicht angegeben, wird ein Datei-Requester geöffnet. Eingabe: [Dateiname] - Name der zu speichernden Datei [FORCE] - verhindert die Ausgabe einer Warnung Ergebnis: -

1.46 Export

Export: Syntax: Export <Format> [Dateiname] [FORCE] Funktion: Speichert die aktuelle Formel als Grafikdatei. Wird [Dateiname] nicht angegeben, wird ein Datei-Requester geöffnet. Eingabe: <Format> - Format der Grafikdatei. In dieser Version ist nur "EPS" verfügbar [Dateiname] - Name der zu speichernden Datei. [FORCE] - verhindert die Ausgabe einer Warnung Ergebnis: -

1.47 Info

Info: Syntax: Info Funktion: Öffnet das gleiche Info-Fenster wie der Menüpunkt **Info ...**. Eingabe: - Ergebnis: -

1.48 version

Version: Syntax: Version Funktion: Liefert die aktuelle Version von MathScript Eingabe: - Ergebnis: result - Versionsstring der Form "MathScript xx.xx"

1.49 screentofront

ScreenToFront: Syntax: ScreenToFront Funktion: Holt den Screen, auf dem sich das Hauptfenster von MathScript befindet, in den Vordergrund. Eingabe: - Ergebnis: -

1.50 requester

Requester: Syntax: Requester [Titel] [Muster] Funktion: Öffnet einen Datei-Requester zum Laden. Eingabe: [Titel] - Titelzeile des Requesters [Muster] - Amiga-Dos-Muster, nach dem Dateien zum Anzeigen ausgewählt werden Ergebnis: -

1.51 Quit

Quit: Syntax: Quit [FORCE] Funktion: Beendet das Programm nach einer Sicherheitsabfrage. Eingabe: [FORCE] - verhindert die Ausgabe der Warnung Ergebnis: -

1.52 getprefs

GetPrefs: Syntax: GetPrefs [PUBSCREEN] [PATH] [NOICONS] [INCLUDE_PS_FONT] Funktion: Liefert die aktuellen Einstellungen Eingabe: [PUBSCREEN] liefert in Ergebnis den Namen des öffentlichen Bildschirms ('def' für den Standard öffentlichen Bildschirm, 'Workbench' für die Workbench) [PATH] liefert in Ergebnis den voreingestellten Pfad [NOICONS] liefert in Ergebnis den Wert YES bzw NO [INCLUDE_PS_FONT] liefert in Ergebnis den Wert YES bzw NO [PS_FONTSIZE] liefert die Höhe des Fonts Ergebnis: liefert die in der Eingabe geforderten Werte Beispiel: GetPrefs PUBSCREEN PATH NOICONS könnte z.B. folgende Zeichenkette in result ablegen: 'Workbench MathScript:Formulas NO'

1.53 setprefs

SetPrefs: Syntax: SetPrefs [PUBSCREEN name] [PATH pfad] [NOICONS YES/NO] [INCLUDE_PS_FONT YES/NO] [PROMPT] Funktion: Ändert die Einstellungen des Programms Eingabe: [PUBSCREEN name] Setzt MathScript auf einen neuen Bildschirm. Mögliche Werte sind: 'def' für den Standard öffentlichen Bildschirm, 'Workbench' für die Workbench oder eine Name

eines geöffneten öffentlichen Bildschirms. [PATH pfad] Setzt den Standard-Pfad auf pfad [NOICONS YES/NO] Bestimmt, ob Piktogramme erstellt werden sollen [INCLUDE_PS_FONT YES/NO] Bestimmt, ob der PostScript-Font beim Exportieren als EPS integriert werden soll [PROMPT] Wird dieses Argument angegeben, werden alle anderen Angaben ignoriert und das **Einstellungs-Fenster** geöffnet Ergebnis: -

1.54 codes

Übersicht aller Steuer-Codes

Positionierung:

Hoch: Syntax: $\langle \text{Obj1} \rangle^{\langle \text{Obj2} \rangle}$ Funktion: Stellt $\langle \text{Obj2} \rangle$ abhängig von der Größe von $\langle \text{Obj1} \rangle$ hoch, um z.B. eine Potenz darzustellen. Eingabe: $\langle \text{Obj1} \rangle$ - Basis $\langle \text{Obj2} \rangle$ - Exponent

Tief: Syntax: <Obj1>_<Obj2> Funktion: Stellt <Obj2> abhängig von der Größe von <Obj1> tief, um z.B. einen Index darzustellen.
Eingabe: <Obj1> - Basis <Obj2> - Index

Hoch und Tief: Syntax: <Obj1>^_<Obj2><Obj3> Funktion: Stellt <Obj2> abhängig von der Größe von <Obj1> hoch und <Obj3> tief. Eingabe: <Obj1> - Basis <Obj2> - Exponent <Obj3> - Index

Oben: Syntax: <Obj>\up<Obj2> Funktion: Positioniert <Obj2> oberhalb von <Obj1>. Eingabe: <Obj1> - Grundobjekt <Obj2>
- oberes Objekt

Beispiel: $-1 \in \mathbb{F}_p \rightarrow \mathbb{F}_p$ (Umkehrfunktion)

Unten: Syntax: $\langle \text{Obj1} \rangle \backslash \text{dn} \langle \text{Obj2} \rangle$ Funktion: Positioniert $\langle \text{Obj2} \rangle$ unterhalb von $\langle \text{Obj1} \rangle$. Eingabe: $\langle \text{Obj1} \rangle$ - Grundobjekt $\langle \text{Obj2} \rangle$ - unteres Objekt Beispiel: $\lim [\lim] \backslash \text{dn} [n \rightarrow 0] \rightarrow n \rightarrow 0$

Oben und Unten: Syntax: <Obj1>\ud<Obj2><Obj3> Funktion: Positioniert <Obj2> oberhalb, <Obj3> unterhalb von <Obj1>.
Eingabe: <Obj1> - Grundobjekt <Obj2> - oberes Objekt <Obj3> - unteres Objekt

Klammern:

Syntax: Eingabe Ausgabe (<Obj1> -> (<Obj1> { <Obj1> -> { <Obj1> \l[<Obj1> -> [<Obj1> \l] <Obj1> ->] <Obj1> \l
<Obj1> -> | <Obj1> <Obj1>) -> <Obj1>) <Obj1> } -> <Obj1> } <Obj1> \r -> <Obj1>] <Obj1> \r[-> <Obj1> [<Obj1> \r[->
<Obj1> | Außerdem gibt es spitze Klammern, die nur über die **Auswahl-gadgets** anwählbar sind: < <Obj1> -> < <Obj1> <Obj1>
> -> <Obj1> >

Funktion: Skaliert die Klammer in Abhängigkeit von der Größe von <Obj1> Eingabe: <Obj1> - Größe der Klammer

Beispiel: $\vdash 1 \mid ([1/2x]r \rightarrow \mid -x \mid \wedge 2 \mid$

Bruch- und Wurzel-Bildung:

Bruch: Syntax: <Obj1>/<Obj2> Funktion: Erzeugt einen Bruch. Eingabe: <Obj1> - Zähler <Obj2> - Nenner

Quadratwurzel: Syntax: `\sqrt{Obj1}` Funktion: Erzeugt eine Quadraturzel Eingabe: `<Obj1>` - Radikand

Wurzel: Syntax: \rt<Obj1><Obj2> Funktion: Erzeugt eine Wurzel Eingabe: <Obj1> - Wurzelexponent <Obj2> - Radikand

Da die Zeichen ``',_','\` und ' / ' als Steuer-Codes verwendet werden, kann man sie nur durch voranstellen eines Backslashes (`\`) als normale Zeichen verwenden. Also: \^ -> ^ \ -> \| -> \| \ / -> /

1.55 Nutzungsbedingungen

MathScript ist Shareware. Es darf unter Beachtung folgender Bedingungen frei kopiert werden :

- Der Verkaufspreis darf nicht höher als der Preis des leeren Datenträgers plus eine Kopiergebühr plus Versandkosten.
- Alle Teile des Programms müssen vollständig sein. Die Weitergabe einzelner Dateien ist nicht erlaubt.
- MathScript oder Teile des Programms dürfen nicht ohne die schriftliche Erlaubnis des Autors zusammen mit kommerzieller Software vertrieben werden.
- Alle Teile des Software-Packets dürfen nicht verändert werden.
- Der Autor ist nicht verantwortlich für Schäden, die durch das Programm verursacht worden sind.

Das Programm darf für 14 Tage benutzt werden. Danach ist für eine weitere Benutzung die **Registrierung** nötig.

1.56 registrierung

Wenn Sie sich registrieren lassen, erhalten Sie die neueste Version von MathScript und ein Key-File. Mit diesem sind Sie dann auch berechtigt, kommende Ausgaben von MathScript zu nutzen. Außerdem entfallen dann auch die "Nerv-Requester".

Die Shareware-Gebühr beträgt 20DM oder 15\$.

Um sich registrieren zu lassen, drucken Sie das **Registrierungs-Formular** aus (oder erstellen Sie ein entsprechendes von Hand) und füllen Sie die leeren Felder mit ihren persönlichen Angaben. Das Ganze schicken Sie dann an:

Simon Ihmig Beim Rauhen Hause 30 22111 Hamburg Germany

1.57 Registrierungs-Formular

Ich möchte mich für das Shareware-Programm MathScript registrieren lassen. Ich erhalte dafür ein Keyfile, das mich zur uneingeschränkten Nutzung des Programms ermächtigt. Ich garantiere, daß ich dieses nur zur eigenen Nutzung gebrauche und nicht weitergebe.

Zahlungsart: (zutreffendes ankreuzen)

☐ Die Gebühr in Höhe von 20DM bzw 15\$ liegt bar bei.

☐ Die Gebühr in Höhe von 20DM bzw 15\$ ist auf folgendes Konto überwiesen: Kontonummer: 1077/782033 Bankleitzahl: 200 505 50 Hamburger Sparkasse Inhaber: Simon Ihmig

Meine Anschrift:

Name: _____

Vorname: _____

Straße: _____

PLZ, Ort: _____

Tel: _____

Land: _____

e-mail: _____

Ort, Datum: _____ Unterschrift: _____

1.58 zukunft

Für zukünftige Versionen von MathScript ist u.a. geplant: (keine Garantie auf Verwirklichung!)

- Export als IFF-ILBM BitMap-Datei
- Einbindung einer TIFF-Grafik in die EPS-Datei zur Vorschauanzeige in Textverarbeitungsprogrammen
- mehr Steuer-Codes (Matrix, Unter-, Überstreichungen...)
- mehr Symbole (Zahlenmengensymbole, geometrische Symbole...)
- mehrzeilige Formeln
- Online-Hilfe ?

Damit ich sonstige Wünsche erfüllen kann, um Anregungen zu geben oder auf Fehler hinzuweisen, schreiben Sie mir BITTE !!!
Meine Adresse finden Sie unter dem Inhaltsverzeichnis.
