

Makros

COLLABORATORS

	<i>TITLE :</i> Makros		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		August 7, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1 Makros	1
1.1 TurboCalc by Michael Friedrich	1
1.2 Allgemeines	2
1.3 Zahlen	2
1.4 Wahrheitswerte	2
1.5 Texte	3
1.6 Zelle/Bereich	3
1.7 Auslassen von Parametern	3
1.8 Auslassen aller Parameter	4
1.9 Rückgabewerte von Befehlen	4
1.10 Menü-Befehle	4
1.11 Blockbefehle	5
1.12 AUSSCHNEIDEN([Block])	5
1.13 BEZUGTYP(Flag[;Block])	5
1.14 BEZUGVERSCHIEBEN(X;Y;Flag;Block)	6
1.15 DATENEINFÜGEN(Modus;[Block])	6
1.16 EINFÜGEN([Block])	6
1.17 ENTFERNEN(Data;[Block])	7
1.18 FÜLLEN(Mode;[Block])	7
1.19 HINZUFÜGEN(Data;[Block])	8
1.20 KOPIEREN([Block])	8
1.21 LÖSCHEN(Data;[Block])	8
1.22 MMAL(Block1;Block2;ErgBlock)	9
1.23 MMINUS(Block1;Block2;ErgBlock)	9
1.24 MPLUS(Block1;Block2;ErgBlock)	9
1.25 REIHEBERECHNEN(Typ;Inkrement;Spalten;Bereich)	10
1.26 SORTIEREN(Aufsteigend;Richtung;Zelle;Bereich)	10
1.27 SPRACHE(Modus;Block)	11
1.28 TRANSPONIEREN([Block])	11
1.29 Formatierungen	11

1.30 ANZEIGEN(Zeile;[Block])	12
1.31 AUSRICHTUNG([Hor];[Vert];[Block])	12
1.32 FARBEN([Farbe1];[Farbe2];[Block])	12
1.33 FIXIEREN(Zelle)	13
1.34 MUSTER(Zahl;[Block])	13
1.35 KASTEN(Links;Rechts;Oben;Unten;[Block])	13
1.36 RAHMEN(Links;Rechts;Oben;Unten;[Block])	14
1.37 SCHRIFT([Num];[Zeichensatz];[Block])	15
1.38 SPALTENBREITE(Breite;[Block])	15
1.39 STDSCHRIFT(Zeichensatz)	16
1.40 STILWECHSELN(Num;[Block])	16
1.41 VERBERGEN(Zeile;[Block])	16
1.42 ZAHLENFORMAT(Format;[Block])	17
1.43 ZEILENHÖHE(Höhe;[Block])	18
1.44 ZELLSCHUTZ([Schreiben];[Formel];[Block])	19
1.45 Cursorsteuerung	19
1.46 AKTIVEZELLE()	20
1.47 AUSWÄHLEN([Block])	20
1.48 INSPALTE(Spalte)	20
1.49 INZEILE(Zeile)	20
1.50 LINKS(Num)	21
1.51 RECHTS(Num)	21
1.52 HOCH(Num)	21
1.53 RUNTER(Num)	21
1.54 SPALTE(Spalte)	22
1.55 LETZTESPALTE()	22
1.56 SUCHEN(Text;Teil;Groß;Spalten;Bereich)	22
1.57 ZEILE(Zeile)	23
1.58 LETZTEZEILE()	23
1.59 Eingabe	23
1.60 AMIGAGUIDE(Datei;Kommando)	23
1.61 befindet).	24
1.62 ANFRAGE(Text[;Titel])	24
1.63 DIALOG(Dialog;Hook)	25
1.64 DLGWEITER(Flag)	27
1.65 EINGABE(Text[;Titel];[Zelle])	27
1.66 HILFE(Num;Datei)	28
1.67 MELDUNG(Text[;Titel])	29
1.68 PAUSE(Zeit)	29

1.69	REQPARA(X;Y;Oktext;Abbruchtext;Zeit)	30
1.70	SCHREIBE(Inhalt[Zelle])	30
1.71	SIGNAL()	31
1.72	Laden/Speichern	31
1.73	ABLAGENLESEN([Block])	32
1.74	ABLAGESCHREIBEN([Block])	32
1.75	BLOCKSPEICHERN([Block];[Name])	32
1.76	CSVBLOCKSPEICHERN([Block];[Name];[Trennzeichen])	32
1.77	CSVEINFÜGEN([Block];[Name];[Trennzeichen])	33
1.78	CSVLADEN([Name];[Trennzeichen])	33
1.79	CSVSPERICHERN([Name];[Trennzeichen])	34
1.80	KONFIGLADEN()	34
1.81	KONFIGSPEICHERN()	34
1.82	LADEN([Name])	35
1.83	PROCALCEINFÜGEN([Block];[Name])	35
1.84	PROCALCLADEN([Name])	35
1.85	SPEICHERN([Name])	36
1.86	SPEICHERNALS([Name])	36
1.87	SYLKBLOCKSPEICHERN([Block];[Name])	36
1.88	SYLKEINFÜGEN([Block];[Name])	37
1.89	SYLKLADEN([Name])	37
1.90	SYLKSPEICHERN([Name])	38
1.91	TCDEINFÜGEN([Block];[Name])	38
1.92	Datenbank	38
1.93	DATENBANK([Bereich])	39
1.94	DBEXTRAHIEREN([Zelle])	39
1.95	DBLETZTER(Zelle)	39
1.96	DBLÖSCHEN()	40
1.97	DBSORTIEREN(Aufsteigend;[Zelle])	40
1.98	DBSUCHEN([Zelle])	40
1.99	KRITERIEN([Bereich])	41
1.100	Optionen	41
1.101	BILDAUFBAU(Modus)	42
1.102	DARSTELLUNG(Titel;Raster;Werkzeu;l;Formeln;Null;Cursorm,..)	42
1.103	DEFPUBSCREEN(Bildschirm)	43
1.104	DRUCKBEREICH(Aktivieren;[Bereich])	43
1.105	DRUCKFORMAT(LR;RR;OR;UR;Stil;KZeile;KText;FZeile;FText;Titel;Raster)	44
1.106	GLOBALEOPTIONEN()	44
1.107	LADEOPTIONEN(KeinAutoMakro;Verbergen;Passwort)	44

1.108LÄNDEROPTIONEN(ZF1;ZF2;DF;Währung;WPräfix;WSuffix;Inch)	45
1.109PAPIERFORMAT()	46
1.110SCHUTZOPTIONEN()	46
1.111SEITENVORSCHUB(Flag)	46
1.112SHANGHAI(Modus)	47
1.113SMARTREFRESH(Flag)	47
1.114TABELLENOPTIONEN(MaxBreite;MaxHöhe;Berechnung;Return;Richtung;Icons)	47
1.115TASTE(Taste;Befehl)	48
1.116ZWISCHENABLAGE(Einheit;Trennzeichen;Anführungszeichen)	49
1.117ZWISCHENSPEICHERN(Flag)	49
1.118Menübefehle	50
1.119LÖSCHEMENÜPUNKT(Titel;Punkt)	50
1.120LÖSCHEMENÜTITEL(Titel)	50
1.121LÖSCHEMENÜUNTER(Titel;Punkt;Unter)	50
1.122NEUERMENÜPUNKT(Name;Befehl;[Titel;Punkt])	51
1.123NEUERMENÜTITEL(Name;[Titel])	51
1.124NEUERMENÜUNTER(Name;Befehl;[Titel;Punkt;Unter])	51
1.125NEUESMENÜ()	52
1.126ZEIGEMENÜ()	52
1.127Steuerung	52
1.128AUSFÜHREN(Datei;Parameter;[Fenster])	53
1.129BEENDEN([Flag])	53
1.130BLOCKNAMENFESTLEGEN(Name;Block)	53
1.131DRUCKEN(Drucker;File;NLQ;Bereich;Seite1;Seite2;LPI;Farbig;Pause;Größe;Breite;Höhe)	54
1.132IFFDRUCKEN(Datei;Breite;Höhe;Tiefe;Block)	55
1.133MAKROABSPIELEN(Zelle)	55
1.134NAMENFESTLEGEN(Name;Wert)	55
1.135NAMENLÖSCHEN(Name)	56
1.136NEUBERECHNEN([Modus])	56
1.137NEUETABELLE(Name)	57
1.138RUN(Datei;Parameter;[Fenster])	57
1.139RX(Datei;Port)	57
1.140START(Dateiname)	58
1.141TABELLEEINBLENDEN(Tabellenname;Fensternummer)	58
1.142TABELLESCHLIEßEN(Sofort)	59
1.143TABELLEVERBERGEN(Tabellenname;Fensternummer)	59
1.144UNGEÄNDERT()	60
1.145WÄHLETABELLE(Name[;FensterNummer])	60
1.146Bildschirm	60

1.147ALTEFARBEN()	61
1.148BILDSCHIRM(Breite;Höhe;Tiefe;Modus)	61
1.149FARBEÄNDERN(Farbe;Rot;Grün;Blau)	62
1.150FENSTERAKTIVIEREN()	62
1.151FENSTERANORDNEN()	62
1.152FENSTERANORDNEN2()	62
1.153FENSTERGRÖßE(Breite;Höhe)	63
1.154FENSTERNACHHINTEN()	63
1.155FENSTERNACHVORN()	63
1.156FENSTERPOSITIONIEREN(X;Y;Breite;Höhe)	63
1.157FENSTERVERSCHIEBEN(X;Y)	63
1.158IKONIFIZIEREN()	63
1.159STDFARBEN()	64
1.160ZEICHENSATZ(Zeichensatz;Modus)	64
1.161 Sonstige Befehle	64
1.162AUFNAHME()	65
1.163COMMAND()	65
1.164DIAEINBLENDEN()	65
1.165DRUCKVORSCHAU()	65
1.166EDIT()	66
1.167FORMELEINFÜGEN()	66
1.168MAKROEINFÜGEN()	66
1.169NAMENEINFÜGEN()	66
1.170NEUESFENSTER()	66
1.171OBJEKT(Name;Typ;Text;Wert1;Wert2;Block)	66
1.172OBJEKTAUSWAHL(Name)	67
1.173OBJEKTKLASSE(Name)	67
1.174OBJEKTPARA(Name;Makro;Makroaktivierung;Farbe;Rahmen)	68
1.175OBJEKTPOS(Name;X;Y;Breite;Höhe)	68
1.176RÜCKGÄNGIG()	68
1.177SPEZIALOPTIONEN(Flags)	68
1.178STOPAUFNAHME()	69
1.179SYSINFO()	69
1.180TCMAKRO(TCLib;Offset;Num1;Num2;Text)	69
1.181ÜBER()	69
1.182WIEDERHERSTELLEN()	69
1.183 Makro-Steuerung	69
1.184BEIFEHLER(ZELLE)	70
1.185BIS(Bedingung)	70

1.186FÜR(Variable;Anfang;Ende;Schritt)	71
1.187FÜRBLOCK(Variable;Block;Flags)	72
1.188MAKRO(...)	72
1.189MSONST()	73
1.190MWENN(Bedingung)	73
1.191MWENNENDE()	73
1.192NÄCHSTES()	74
1.193ROUTINE(Zelle)	74
1.194RÜCKSPRUNG([Zelle])	74
1.195SCHLEIFE()	74
1.196SCHRITT([Flag])	75
1.197SOLANGE(Bedingung)	76
1.198SPRINGE(Zelle)	76
1.199SPRINGEWENN(Bedingung;Zelle)	76
1.200spezielle ARexx-Befehle	77
1.201KOMMENTAR	77
1.202LIESWERT [Wert]	77
1.203LIESFORMEL [Zelle]	78
1.204LIESCURSORPOS	78
1.205Komplettes Inhaltsverzeichnis	79
1.206Index	84

Chapter 1

Makros

1.1 TurboCalc by Michael Friedrich

TurboCalc - copyright Michael Friedrich.

Komplettes [Inhaltsverzeichnis](#) dieser Datei.

Haupt-Inhaltsverzeichnis

Kompletter Index (über alle Dateien)

Befehle

[Allgemeines](#)

[Blockbefehle](#)

[Formatierungen](#)

[Cursorsteuerung](#)

[Eingabe](#)

[Laden/Speichern](#)

[Datenbank](#)

[Optionen](#)

[Menübefehle](#)

[Steuerung](#)

[Bildschirm](#)

[Sonstige Befehle](#)

[Makro-Steuerung](#)

[spezielle ARexx-Befehle](#)

Dies ist die gemeinsame Befehlsübersicht für die ARexx- und MAKRO-Befehle. Einzelheiten zu beiden Sprachen entnehmen Sie bitte dem Kapitel "Makro/ARexx".

Der Befehle bestehen normalerweise aus einem Befehlswort gefolgt von Parametern (durch Semikolon getrennt) in Klammern (etwa LÖSCHEN(A1:C5)). Benötigt ein Befehl keine Parameter, so können die Klammern auch entfallen.

Weiterhin ist auch (jedoch nur für den ARexx-Modus) eine Schreibweise ohne Klammern möglich, in der die Parameter durch Leerzeichen getrennt werden (wie im ARexx-Modus üblich), es ist also folgendes möglich:

FÜLLEN(0;A1:C5)

FÜLLEN 0 A1:C5

FÜLLEN(0)

FÜLLEN 0

(warum FÜLLEN sowohl einen als auch zwei Parameter besitzen kann, können Sie bei FÜLLEN nachlesen).

(In der Schreibweise ohne Klammern können Texte ohne Anführungszeichen eingegeben werden, falls sie keine Leerzeichen oder Anführungszeichen enthalten - bei der Klammerschreibweise sind sie nötig!)

MELDUNG("Dies ist eine Meldung";"Titel")

MELDUNG "Dies ist eine Meldung" "Titel"

MELDUNG Meldung Titel

1.2 Allgemeines

Allgemeines

Vor der eigentlichen Befehlsübersicht noch einige Hinweise, die sich auf fast alle Befehle beziehen:

Viele Befehle beziehen sich auf Menüpunkte gleichen oder ähnlichen Namens, hier wurde dann auf eine ausführliche Beschreibung des Befehls verzichtet - sollten Ihnen bei der Beschreibung des Befehls noch etwas unklar sein, so schauen Sie doch einfach bei der Beschreibung des entsprechenden Menüpunktes nach.

Die meisten Befehlen verfügen über folgende Standard-Parameter:

Zahlen

Wahrheitswerte

Texte

Zelle/Bereich

Auslassen von Parametern

Auslassen aller Parameter

Rückgabewerte von Befehlen

Menü-Befehle

1.3 Zahlen

Zahlen

(Etwa bei FARBEN). Hierbei handelt es sich meist um ganze Zahlen, die Grenzen entnehmen sie den Funktionen. Sie können diese sowohl als Ganzzahl als auch als Fließkommazahl oder als Formeln, die einen dieser Werte zurückgeben, angeben.

1.4 Wahrheitswerte

Wahrheitswerte

Diese werden häufig für Ja/Nein-Entscheidungen benutzt. (Etwa bei DBSORTIEREN für auf/absteigende Sortierung). Hier entspricht dann der Wert 0 bzw. FALSCH einem Nein und alles andere ergibt Ja. (Wird dieser Parameter weggelassen, so entspricht dies entweder Ja oder aber dieses Flag wird ignoriert, mehr dazu siehe bei den jeweiligen Funktionen)

1.5 Texte

Texte

Hier können Sie beliebige Texte eingeben, diese müssen jedoch immer in Anführungszeichen stehen (außer im ARexx-Format ohne Klammern). Außerdem können natürlich beliebige Text-Funktionen benutzt werden.

Hinweis: Genau wie bei den Funktionen müssen im Text vorkommende Anführungszeichen verdoppelt werden: der Text a"b muß als "a""b" und der Text a""b als a""""b geschrieben werden.

bei ARexx gelten einige Besonderheiten:

Diese Texte müssen normalerweise nicht von Anführungszeichen umgeben werden, außer es sind Leerzeichen enthalten.

Dann müssen auch hier Leerzeichen folgen, aufgrund einer Besonderheit von ARexx (da Leerzeichen Parameter abtrennen) müssen diese Texte dabei sowohl in doppelte als auch in einfache (außen herum) Anführungszeichen eingeschlossen werden, etwa:

```
MELDUNG ``Dies ist ein Text``
```

oder der gesamte Befehl wird mit Hochkommata umschlossen (dies ist meist übersichtlicher!):

```
'MELDUNG "Dies ist ein Text"'
```

1.6 Zelle/Bereich

Zelle/Bereich

Dies ist der letzte Parameter von fast allen Funktionen und gibt dann an, auf welchen Bereich diese Funktion angewendet werden soll - wird er weggelassen, so wird die aktuelle Zelle bzw. der aktuelle Block benutzt. Allgemein gilt jedoch folgendes:

Bei Bereich kann es sich um eine beliebige Zellangabe (etwa C5 oder C13:F25) handeln. Auch die Funktionen ZELLE bzw. ZELLEABS sowie die AT-Funktion (@Tabelle;Bereich) sind möglich.

Tip: Mit der AT-Funktion (genaue Beschreibung siehe Zell-Funktionen) kann der Makrobefehl auf beliebige geladene Tabellen angewendet werden, ohne daß diese zuvor extra aktiviert werden muß. So ändert FARBEN(3;;@Datenbank:C5) die Farbe der Zelle C5 der Tabelle mit dem Namen "Datenbank".

1.7 Auslassen von Parametern

Auslassen von Parametern

Viele Befehle enthalten Parameter die angegeben werden können, jedoch nicht angegeben werden müssen. In diesem Fall kann man die entsprechenden Parameter auch weglassen (und einfach das abschließende Semikolon bzw. Klammer schreiben). (Folgen darauf keine Parameter mehr bzw. möchte man diese auch nicht angeben, so kann man alle Semikolon zwischen dem letzten Parameter und der abschließenden Klammer ")" weglassen.)

Beispiel: Statt FARBEN(3;4;A1:C5) kann man auch FARBEN(3;;A1:C5) schreiben. Dann wird die Hintergrundfarbe nicht verändert. Oder man schreibt FARBEN(3), dann wird die aktuelle Zelle bzw. der aktuelle Block benutzt.

Hinweis: Aus diesem Grund ist bei fast jedem Befehl der letzte Parameter eine Blockangabe. Sie kann weggelassen werden, dann bezieht sich der Befehl auf den aktuellen Block - wird sie jedoch angegeben, so stellt dies der gewünschte Bereich dar.

Bei ARexx kann das Auslassen von Parametern dadurch erzeugt werden, daß statt des Parameters ein einfaches "_" bzw. # (tiefes Minus bzw. Doppelkreuz) gesetzt wird (steht ein Parameter am Ende des Befehls, so kann er auch komplett weggelassen werden).

```
FARBEN 3 _ C5
```

```
FARBEN 3 # C5
```

```
FARBEN 3
```

1.8 Auslassen aller Parameter

Auslassen aller Parameter

Bei vielen Befehlen können auch alle Parameter ausgelassen werden - es erscheint dann der jeweilige Requester, mit dem die Daten vom Benutzer selbst eingestellt werden können. (Etwa: FARBEN() zum Anzeigen des Farbauswahl-Fensters.)

Achtung: Diese Befehle sind in Makro und ARexx-Programm nur mit folgender Einschränkung einzusetzen: Da TurboCalc für die Bildschirmanzeige und Fensterverwaltung einen eigenen Task startet und so ein multitaskingfähiges Arbeiten ermöglicht (so können Fenster verschoben werden, selbst wenn z.B. gerade im Hintergrund sortiert... wird), muß bei diesen Befehlen die Kontrolle an den anderen Task übergeben werden. Deswegen muß das Makro bzw. der ARexx-Befehl auch nicht warten, bis der Benutzer >OK< bzw. >Abbruch< drückt, sondern der Befehl wird sofort mit dem Status OK beendet. Das Makro wird also beim nächsten Befehl fortgesetzt, obwohl das Fenster noch zu sehen ist. Achten Sie bei der Programmierung ggf. darauf.

1.9 Rückgabewerte von Befehlen

Rückgabewerte von Befehlen

Einige Befehle können auch scheitern (etwa LADEN oder TABELLEAUSWÄHLEN). Im Direktmodus erhalten Sie dann eine entsprechende Fehlermeldung.

Wird der entsprechende Befehl jedoch im Makromodus aufgerufen, so wird diese Fehlermeldung unterbunden und stattdessen ein Rückgabewert zurückgegeben. So kann der Erfolg der Operation geprüft werden, ohne daß der Benutzer des Makros Fehlermeldungen erhält. In diesem Fall wird der Rückgabewert in die Zelle geschrieben, in dem sich der Makrobefehl befindet. Dort kann der Rückgabewert dann normal abgefragt und ggf. entsprechend verzweigt werden. Normalerweise wird bei Erfolg WAHR zurückgegeben und ansonsten FALSCH, Einzelheiten entnehmen Sie jedoch den einzelnen Befehlen.

Hier ein einfaches Beispiel - in einer realen Anwendung sollte das "Konnte Datei nicht öffnen" natürlich etwas ausführlicher sein (welche Datei und was kann dagegen getan werden..), damit der Benutzer vor dem Rechner nicht verzweifelt.

```
A10 =LADEN("Beispiel.TCD")
```

```
A11 =SPRINGEWENN(#A10;A14)
```

```
A12 =MELDUNG("Konnte Datei nicht öffnen!")
```

```
A13 =SPRINGE(A30).
```

```
A14 hier beginnt das eigentliche Programm
```

```
...
```

```
A30 =RETURN()
```

Beachten Sie bitte das "#" vor der Zelle A10. Dies stellt sicher, daß auch wirklich auf die Zelle des Makroblattes zugegriffen wird und nicht das aktuelle Arbeitsblatt benutzt wird.

1.10 Menü-Befehle

Menü-Befehle

Einige (wenige) Befehlsbeschreibungen beschränken sich auf "Dies entspricht ...", diese Befehle rufen nur den entsprechenden Menüpunkt auf, ggf. erscheint dann das entsprechende Einstellfenster (etwa AUFNAHME, um die Makroaufnahme zu starten). Diese Befehle haben für die "Makro-Programmierung" fast keine praktische Bedeutung. Sie sind hauptsächlich zur Menü- oder Tastendefinition gedacht, etwa:

```
NEUERMENÜPUNKT("Aufname";"AUFNAME")
```

```
TASTE("CTRL-A";"AUFNAHME")
```

1.11 Blockbefehle

Blockbefehle

AUSSCHNEIDEN([Block])

BEZUGTYP(Flag[;Block])

BEZUGVERSCHIEBEN(X;Y;Flag;Block)

DATENEINFÜGEN(Modus;[Block])

EINFÜGEN([Block])

ENTFERNEN(Data;[Block])

FÜLLEN(Mode;[Block])

HINZUFÜGEN(Data;[Block])

KOPIEREN([Block])

LÖSCHEN(Data;[Block])

MMAL(Block1;Block2;ErgBlock)

MMINUS(Block1;Block2;ErgBlock)

MPLUS(Block1;Block2;ErgBlock)

REIHEBERECHNEN(Typ;Inkrement;Spalten;Bereich)

SORTIEREN(Aufsteigend;Richtung;Zelle;Bereich)

SPRACHE(Modus;Block)

TRANSPONIEREN([Block])

1.12 AUSSCHNEIDEN([Block])

AUSSCHNEIDEN([Block])

(Englisch: CUT)

Schneidet den aktuellen Block bzw. den durch Block angegebenen aus (d.h. leert die Zellen und speichert den Block für EINFÜGEN intern zwischen).

Block: bestimmt den Block. Fehlt diese Angabe, so wird der aktuelle Block angenommen.

Dies entspricht (ohne Parameter) <Bearbeiten-Ausschneiden>.

Verwandte Befehle:

KOPIEREN , **EINFÜGEN** , **DATENEINFÜGEN**

1.13 BEZUGTYP(Flag[;Block])

BEZUGTYP(Flag[;Block])

(Englisch: REFERENCETYPE)

Wandelt alle Bezüge (etwa C5 oder \$C\$5) der Formeln innerhalb Block entsprechend Flag um:

0 alle Bezüge in absolut (\$C\$5) umwandeln

1 alle Bezüge in relativ (C5) umwandeln

2 alle Bezüge tauchen (Standard, falls dieser Parameter ausgelassen wird)

Block: Der Block, aus dem die Formeln überprüft und geändert werden sollen. Wird dieser Parameter ausgelassen, so wird der aktuell markierte Block verwandt.

Ohne Parameter entspricht dies <Befehle-Bezüge-Rel->Abs>, dort finden Sie auch eine weitere Beschreibung.

1.14 BEZUGVERSCHIEBEN(X;Y;Flag;Block)

BEZUGVERSCHIEBEN(X;Y;Flag;Block)

(Englisch: REFERENCEMOVE)

Verschiebt alle Bezüge der Formeln innerhalb Block um ein bestimmtes Offset.

X: Anzahl der Spalten, um die die Formeln verschoben werden sollen (Positive Werte: nach rechts, negative Werte: nach links)

Y: Anzahl der Zeilen, um die die Formeln verschoben werden sollen (Positive Werte: nach unten, negative Werte: nach oben)

Flag: Bestimmt, welche Formeln verschoben werden sollen:

0 alle Bezüge (Standard, falls der Parameter ausgelassen wird)

1 relative Bezüge

2 absolute Bezüge

Block: Der Block, aus dem die Formeln überprüft und geändert werden sollen. Wird dieser Parameter ausgelassen, so wird der aktuell markierte Block verwendet.

Ohne Parameter entspricht dies <Befehle-Bezüge-Verschieben>, dort finden Sie auch eine weitere Beschreibung.

1.15 DATENEINFÜGEN(Modus;[Block])

DATENEINFÜGEN(Modus;[Block])

(Englisch: PASTEDATA)

Fügt den mit AUSSCHNEIDEN oder KOPIEREN in den Zwischenspeicher kopierten Block an der aktuellen Stelle/Block bzw. wie durch Block angegeben ein. Ist ein Block markiert (bzw. Block angegeben), so wird dieser gefüllt (also das Kopierte entweder wiederholt oder abgeschnitten)

Modus: Dies bestimmt, welche Daten eingefügt werden sollen:

0: Es wird nur das Format eingefügt.

1: Es werden nur die Werte eingefügt, Formeln werden verworfen.

2: Es werden Werte und Formeln eingefügt, das Format jedoch ignoriert.

Block: Bestimmt den Block (oder eine Zelle), an der der Zwischenspeicher eingefügt werden soll. Fehlt er, so wird die aktuelle Cursorposition bzw. der aktuelle Block verwendet.

Dies entspricht (ohne Parameter) <Bearbeiten-Einfügen nur-...>.

Hinweis: Sollen die Daten komplett eingefügt werden, dann muß EINFÜGEN benutzt werden.

Verwandte Befehle:

AUSSCHNEIDEN , **KOPIEREN** , **EINFÜGEN**

1.16 EINFÜGEN([Block])

EINFÜGEN([Block])

(Englisch: PASTE)

Fügt den mit AUSSCHNEIDEN oder KOPIEREN in den Zwischenspeicher kopierten Block an der aktuellen Stelle/Block bzw. wie durch Block angegeben ein. Ist ein Block markiert (bzw. Block angegeben), so wird dieser gefüllt (also das Kopierte entweder wiederholt oder abgeschnitten)

Block: Bestimmt den Block (oder eine Zelle), an der der Zwischenspeicher eingefügt werden soll. Fehlt er, so wird die aktuelle Cursorposition bzw. der aktuelle Block verwendet.

Dies entspricht (ohne Parameter) <Bearbeiten-Einfügen>.

Hinweis: Soll entweder nur das Format oder nur der Inhalt eingefügt werden, so muß dazu DATENEINFÜGEN benutzt werden.

Verwandte Befehle:

AUSSCHNEIDEN , **KOPIEREN** , **DATENEINFÜGEN**

1.17 ENTFERNEN(Data;[Block])

ENTFERNEN(Data;[Block])

(Englisch: REMOVE)

Entfernt von dem aktuellen bzw. angegebenen Block alle Informationen und verschiebt den Rest nach links/oben bzw. entfernt die ganzen Zeilen/Spalten, und zwar:

Data: Bestimmt, wie gelöscht werden soll:

0: Die ganze Spalte wird gelöscht

1: Die ganze Zeile wird gelöscht

2: Der Block wird gelöscht und der Rest (rechts des Blockes) nach links geschoben

3: Der Block wird gelöscht und der Rest (unter dem Block) nach oben geschoben

Bei allen anderen Werten erscheint eine Fehlermeldung!

Block: Bestimmt den Block, bei dem etwas gelöscht werden soll. Fehlt diese Angabe, so wird der aktuelle Block bzw. die aktuelle Zelle benutzt.

Wird kein Parameter angegeben, so entspricht dies <Bearbeiten-Zellen entfernen...> - es wird das entsprechende Fenster zur Auswahl angezeigt. (Weitere Hinweise hierzu siehe bei "Makro-Befehle" in der Einleitung!)

Verwandte Befehle:

AUSSCHNEIDEN , **LÖSCHEN** , **HINZUFÜGEN**

1.18 FÜLLEN(Mode;[Block])

FÜLLEN(Mode;[Block])

(Englisch: FILL)

Füllt den aktuellen bzw. angegebenen Block wie folgt:

Mode: Bestimmt, wie gefüllt werden soll:

0: Rechts - In jeder Zeile (des Blocks) wird der Wert ganz links in alle anderen Zellen kopiert

1: Unten - Die obersten Werte werden spaltenweise nach unten kopiert.

2: Links - Von ganz rechts werden die Werte in alle Zellen rechts davon kopiert.

3: Oben - Die unterste Zeile bestimmt das Füllmuster.

Bei allen anderen Werten erscheint eine Fehlermeldung!

Block: Bestimmt den Block, der gefüllt werden soll. Fehlt diese Angabe, so wird der aktuelle Block bzw. die aktuelle Zelle benutzt.

Dies entspricht den Menüpunkten <Bearbeiten-Füllen-...>

Verwandte Befehle:

KOPIEREN und **EINFÜGEN** , **NEUBERECHNEN**

1.19 HINZUFÜGEN(Data;[Block])

HINZUFÜGEN(Data;[Block])

(Englisch: ADD)

Fügt einen Block in der Größe des aktuellen bzw. angegebenen Blocks an dessen linken oberen Ecke hinzu und verschiebt den alten Inhalt nach rechts/unten bzw. fügt ganzen Zeilen/Spalten ein, und zwar:

Data: Bestimmt, wie eingefügt werden soll:

0: Eine (mehrere) ganze Spalte wird eingefügt.

1: Eine (mehrere) ganze Zeile wird eingefügt.

2: Der Block wird eingefügt, und der alte Inhalt (links des Blocks) nach rechts geschoben

3: Der Block wird eingefügt, und der alte Inhalt (über dem Block) nach unten geschoben

Bei allen anderen Werten erscheint eine Fehlermeldung!

Block: Bestimmt den Block, der eingefügt werden soll. Fehlt diese Angabe, so wird der aktuelle Block bzw. die aktuelle Zelle benutzt.

Wird kein Parameter mit angegeben, so entspricht dies <Bearbeiten-Zellen hinzufügen...> - es wird das entsprechende Fenster zur Auswahl angezeigt. (Weitere Hinweise hierzu siehe bei "Makro-Befehle" in der Einleitung!)

Verwandte Befehle:

AUSSCHNEIDEN , LÖSCHEN , ENTFERNEN

1.20 KOPIEREN([Block])

KOPIEREN([Block])

(Englisch: COPY)

Kopiert den aktuellen bzw. den durch Block angegebenen Block in den internen Zwischenspeicher (für EINFÜGEN).

Block: bestimmt den Block. Fehlt diese Angabe, so wird der aktuelle Block angenommen.

Dies entspricht (ohne Parameter) <Bearbeiten-Kopieren> .

Verwandte Befehle:

AUSSCHNEIDEN , EINFÜGEN , DATENEINFÜGEN

1.21 LÖSCHEN(Data;[Block])

LÖSCHEN(Data;[Block])

(Englisch: CLEAR)

Löscht von dem aktuellen bzw. angegebenen Block folgende Informationen:

Data: Bestimmt, welche Informationen gelöscht werden sollen: 0=Alles, 1=Format, 2=Formeln (Werte der letzten Berechnung bleiben erhalten), 3=Werte und Formeln (Formatierung bleibt erhalten). Alle anderen Werte (und auch auslassen des Parameters) ergeben eine Fehlermeldung!

Block: Bestimmt den Block, bei dem etwas gelöscht werden soll. Fehlt diese Angabe, so wird der aktuelle Block bzw. die aktuelle Zelle benutzt.

Wird kein Parameter angegeben, so entspricht dies <Bearbeiten-Löschen> - es wird das entsprechende Fenster zur Auswahl angezeigt. (Weitere Hinweise hierzu siehe bei "Makro-Befehle" in der Einleitung!)

Verwandte Befehle:

AUSSCHNEIDEN , ENTFERNEN , HINZUFÜGEN

1.22 MMAL(Block1;Block2;ErgBlock)

MMAL(Block1;Block2;ErgBlock)

(Englisch: MMUL)

Dies ist ein Matrix-Rechenbefehl. Er berechnet das Produkt zweier Matrizen ($\text{ErgBlock} = \text{Block1} * \text{Block2}$).

In TurboCalc wird eine Matrix als normaler Block aufgefaßt, die einzelnen Elemente einer Matrix entsprechen den jeweiligen Zellen des Blocks.

Block1,Block2: Sind die Quellblöcke. Diese müssen kompatibleGrößen haben (d.h. die Breite des ersten Blockes muß gleich der Höhe des zweiten Blockes sein.) Passen Block1 & Block2 nicht zusammen, so wird eine Fehlermeldung zurückgegeben.

ErgBlock: Hier ist nur die linke obere Ecke wichtig, die Größe des Blockes ergibt sich aus Block1 und Block2 (und natürlich der Operation) - die Größe von ErgBlock wird ignoriert.

Beispiel

siehe Beispieltabelle "Matrizen.TCD".

Verwandte Befehle

MMINUS , **MPLUS**

1.23 MMINUS(Block1;Block2;ErgBlock)

MMINUS(Block1;Block2;ErgBlock)

(Englisch: MSUB)

Dies ist ein Matrix-Rechenbefehl. Er berechnet die Differenz zweier Matrizen ($\text{ErgBlock} = \text{Block1} - \text{Block2}$).

In TurboCalc wird eine Matrix als normaler Block aufgefaßt, die einzelnen Elemente einer Matrix entsprechen den jeweiligen Zellen des Blocks.

Block1,Block2: Sind die Quellblöcke. Diese müssen kompatibleGrößen haben (d.h. die Blöcke müssen gleich groß sein) Passen Block1 & Block2 nicht zusammen, so wird eine Fehlermeldung zurückgegeben.

ErgBlock: Hier ist nur die linke obere Ecke wichtig, die Größe des Blockes ergibt sich aus Block1 und Block2 (und natürlich der Operation) - die Größe von ErgBlock wird ignoriert.

Beispiel

siehe Beispieltabelle "Matrizen.TCD".

Verwandte Befehle

MMAL , **MPLUS**

1.24 MPLUS(Block1;Block2;ErgBlock)

MPLUS(Block1;Block2;ErgBlock)

(Englisch: MADD)

Dies ist ein Matrix-Rechenbefehl. Er berechnet die Summe zweier Matrizen ($\text{ErgBlock} = \text{Block1} + \text{Block2}$).

In TurboCalc wird eine Matrix als normaler Block aufgefaßt, die einzelnen Elemente einer Matrix entsprechen den jeweiligen Zellen des Blocks.

Block1,Block2: Sind die Quellblöcke. Diese müssen kompatibleGrößen haben (d.h. die Blöcke müssen gleich groß sein) Passen Block1 & Block2 nicht zusammen, so wird eine Fehlermeldung zurückgegeben.

ErgBlock: Hier ist nur die linke obere Ecke wichtig, die Größe des Blockes ergibt sich aus Block1 und Block2 (und natürlich der Operation) - die Größe von ErgBlock wird ignoriert.

Beispiel

siehe Beispieltabelle "Matrizen.TCD".

Verwandte Befehle

MMAL , **MMINUS**

1.25 REIHEBERECHNEN(Typ;Inkrement;Spalten;Bereich)

REIHEBERECHNEN(Typ;Inkrement;Spalten;Bereich)

(Englisch: SERIES)

Typ: Gibt den Typ und die Methode an, nach der die Reihenerstellung stattfinden soll:

0: arithmetisch (Standard, falls dieser Parameter ausgelassen wird)

1: geometrisch

2: Tag

3: Wochentag

4: Monat

5: Jahr

Inkrement: Gibt den Wert an, um den der Startwert erhöht werden soll.

Spalten: Die Reihen werden spaltenweise erstellt, falls Spalte WAHR bzw. ungleich Null ist. Bei 0 oder FALSCH werden die Werte in den Zeilen erzeugt.

Bereich: Bestimmt den Bereich, der erstellt werden soll. Fehlt dieser Parameter, so wird der aktuelle Block benutzt.

Ohne Parameter entspricht dieser Befehl dem Menüpunkt <Daten-Reihe berechnen> - es wird dann das entsprechende Fenster zur Einstellung der Parameter geöffnet.

1.26 SORTIEREN(Aufsteigend;Richtung;Zelle;Bereich)

SORTIEREN(Aufsteigend;Richtung;Zelle;Bereich)

(Englisch: SORT)

Hiermit kann ein Bereich sortiert werden und zwar mit folgenden Einstellungen:

Aufsteigend: Sortiert die Daten aufsteigend, falls dieser Parameter ungleich 0 ist oder fehlt. Bei 0 bzw. FALSCH wird absteigend sortiert.

Richtung: Sortiert nach Zeilen (<>0 oder weglassen) oder Spalten (0 bzw. FALSCH).

Zelle: Bestimmt, nach welcher Zeile oder Spalte (je nach Richtung) sortiert werden soll. (Wird dieser Parameter ausgelassen, so wird nach der ersten Zeile oder Spalte sortiert)

Block: Bestimmt den Block, der sortiert werden soll.. Fehlt diese Angabe, so wird der aktuelle Block bzw. die aktuelle Zelle benutzt.

Ohne Parameter entspricht dies dem Menüpunkt <Daten-Sortieren> - es erscheint dann das entsprechende Fenster, in dem man die Einstellungen angeben kann.

1.27 SPRACHE(Modus;Block)

SPRACHE(Modus;Block)

(Englisch: LANGUAGE)

Dieser Befehl entspricht <Befehle-Formel übersetzen-...> und wandelt die Formeln und Makrobefehle des Blocks Block in die gewünschte Sprache um:

Modus: 0=Deutsch, 1=Englisch

Block: Bestimmt den Block, der geändert werden soll. Fehlt diese Angabe, so wird der aktuelle Block bzw. die aktuelle Zelle benutzt.

1.28 TRANSPONIEREN([Block])

TRANSPONIEREN([Block])

(Englisch: TRANSPOSE)

Transponiert den Block Block bzw. den aktuellen Block, d.h. die Zellen werden an der Diagonalen von links oben nach rechts unten gespiegelt. (Die Elemente auf der Diagonalen bleiben unverändert).

Dies entspricht der Methode wie auch Matrizen transponiert werden.

Block: Bestimmt den Block, der transponiert werden soll. Fehlt diese Angabe, so wird der aktuelle Block benutzt.

Hinweis: Der Block muß quadratisch sein (d.h. die Höhe gleich der Breite) - ansonsten ist das Transponieren nicht möglich, und es wird eine entsprechende Fehlermeldung zurückgegeben.

Dies entspricht dem Menüpunkt <Befehle-Transponieren>.

1.29 Formatierungen

Formatierungen

ANZEIGEN(Zeile;[Block])

AUSRICHTUNG([Hor];[Vert];[Block])

FARBEN([Farbe1];[Farbe2];[Block])

FIXIEREN(Zelle)

MUSTER(Zahl;[Block])

KASTEN(Links;Rechts;Oben;Unten;[Block])

RAHMEN(Links;Rechts;Oben;Unten;[Block])

SCHRIFT([Num];[Zeichensatz];[Block])

SPALTENBREITE(Breite;[Block])

STDSCHRIFT(Zeichensatz)

STILWECHSELN(Num;[Block])

VERBERGEN(Zeile;[Block])

ZAHLENFORMAT(Format;[Block])

ZEILENHÖHE(Höhe;[Block])

ZELLSCHUTZ([Schreiben];[Formel];[Block])

1.30 ANZEIGEN(Zeile;[Block])

ANZEIGEN(Zeile;[Block])

(Englisch: SHOW)

Zeigt verborgene Zeilen bzw. Spalten wieder an:

Zeile: Bestimmt, ob die Zeile bzw. die Spalte wieder angezeigt werden soll:

0: Spalte

1: Zeile

Block: Bestimmt den Block, auf den der Befehl angewendet werden soll (Je nach Wert von Zeile wird nur die Zeile bzw. Spalte verwandt). Fehlt diese Angabe, so wird der aktuelle Block bzw. die aktuelle Zelle benutzt.

Dies entspricht den Menüpunkten <Format-Anzeigen-Spalte> bzw. <Format-Anzeigen-Zeile>.

Beispiel:

ANZEIGEN(0) blendet die aktuelle Spalte wieder ein.

SHOW(1;C2:C3) blendet die Zeilen 2 und 3 ein.

1.31 AUSRICHTUNG([Hor];[Vert];[Block])

AUSRICHTUNG([Hor];[Vert];[Block])

(Englisch: ALIGNMENT)

Bestimmt die Ausrichtung des aktuellen bzw. mit Block angegebenen Blocks und zwar:

Hor: 0=Standard, 1=Links, 2=Rechts, 3=Zentriert (alle anderen Werte bzw. weglassen des Parameters verändern die horizontale Ausrichtung nicht.)

Vert: 0=Standard, 1=Oben, 2=Mitte, 3=Unten (alle anderen Werte bzw. weglassen des Parameters verändern die vertikale Ausrichtung nicht.)

Block: bestimmt den Block, für den die Formatierung geändert werden soll. Fehlt diese Angabe, so wird der aktuelle Block angenommen.

Wird kein Parameter angegeben, so entspricht dies <Format-Ausrichtung> - es wird das entsprechende Fenster zur Auswahl angezeigt. (Weitere Hinweise hierzu siehe bei "Makro-Befehle" in der Einleitung!)

Beispiel:

AUSRICHTUNG(3) Zentriert alle Zellen des aktuellen Blocks

AUSRICHTUNG(;1;A1:A10) Die Inhalte der Zellen A1 bis A10 werden nach "oben" gesetzt.

AUSRICHTUNG() es erscheint ein Fenster, in dem der Benutzer die Formatierung auswählen kann.

Verwandte Befehle:

SCHRIFT , **ZAHLENFORMAT** , **FARBEN** , **RAHMEN**

1.32 FARBEN([Farbe1];[Farbe2];[Block])

FARBEN([Farbe1];[Farbe2];[Block])

(Englisch: COLORS)

Bestimmt die Text- und Hintergrundfarbe des aktuellen bzw. mit BLOCK angegebenen Blocks und zwar:

Farbe1: Bestimmt die Textfarbe, 0=Std, 1-... sind die weiteren Farben, Skala siehe "Format-Farben". (-1 oder weglassen dieses Parameters beläßt die Farbe)

Farbe2: Bestimmt die Hintergrundfarbe, 0=Std, 1-... sind die weiteren Farben, Skala siehe <Format-Farben>. (-1 oder weglassen dieses Parameters beläßt die Farbe).

Block: bestimmt den Block, für den die Formatierung geändert werden soll. Fehlt diese Angabe, so wird der aktuelle Block angenommen.

Wird kein Parameter angegeben, so entspricht dies <Format-Farben> - es wird das entsprechende Fenster zur Auswahl angezeigt. (Weitere Hinweise hierzu siehe bei "Makro-Befehle" in der Einleitung!)

Beispiel:

FARBEN(3;A1:A10) Die Inhalte der Zellen A1 bis A10 werden blau (bei den Standardfarben)

FARBEN(;3) bestimmt eine neue Hintergrundfarbe für die aktuelle Zelle.

Verwandte Befehle:

SCHRIFT , **STILWECHSELN** , **AUSRICHTUNG** , **ZAHLENFORMAT** , **RAHMEN**

1.33 FIXIEREN(Zelle)

FIXIEREN(Zelle)

(Englisch: FREEZE)

Fixiert die Zeilen über und die Spalten links der Zelle.

Zelle: Ist die Zelle nach der der Titelbereich festgelegt wird. Wird dieser Parameter ausgelassen, so wird die aktuelle Zelle verwandt.

Ohne Parameter entspricht dieser Befehl also <Format-Fixierung festlegen>. Einzelheiten finden Sie bei dessen Beschreibung.

Beispiel:

FIXIEREN(C2) - Spalte A und B und Zeile 1 werden fixiert und somit eine Art Titel.

FREEZE(A1) - Fixierung wieder aufheben.

1.34 MUSTER(Zahl;[Block])

MUSTER(Zahl;[Block])

(Englisch: PATTERN)

Bestimmt das Hintergrundmuster des aktuellen bzw. mit Block angegebenen Blocks.

Der Parameter Zahl kann Werte zwischen 0 und 15 annehmen und bestimmt eines von 15 möglichen Hintergrundmustern. Der optionale Parameter Block bestimmt den Bereich, für den das Hintergrundmuster geändert werden soll.

Verwandte Befehle:

FARBEN

1.35 KASTEN(Links;Rechts;Oben;Unten;[Block])

KASTEN(Links;Rechts;Oben;Unten;[Block])

(Englisch: BOX)

Bestimmt den Rahmen um den aktuellen bzw. mit Block angegebenen Blockes und zwar:

Links, Rechts, Oben, Unten: Geben die Rahmenstärke für den jeweiligen Rand an:

0: aus (oder auslassen des Parameters)

1: dünn

2: mittel

3: dick

Block: bestimmt den Block, für den die Formatierung geändert werden soll. Fehlt diese Angabe, so wird der aktuelle Block angenommen.

Diese Funktion entspricht RAHMEN, es wird jedoch nur die jeweilige Formatierung am Rand des Blocks geändert.

Beispiel:

KASTEN(;;3;3;A10:C20) Der Block A10:C20 erhält einen oberen und unteren Rand (der Rest bleibt unverändert)

BOX(1;1;1;1;A1:A10) Die Zellen A1 bis A10 erhalten einen dünnen Rand auf allen Seiten.

Verwandte Befehle:

RAHMEN , **FARBEN** , **SCHRIFT** , **STILWECHSELN** , **AUSRICHTUNG** , **ZAHLENFORMAT**

1.36 RAHMEN(Links;Rechts;Oben;Unten;[Block])

RAHMEN(Links;Rechts;Oben;Unten;[Block])

(Englisch: FRAME)

Bestimmt den Rahmen des aktuellen bzw. mit Block angegebenen Blocks und zwar:

Links, Rechts, Oben, Unten: Geben die Rahmenstärke für den jeweiligen Rand an:

0: aus

1: dünn

2: mittel

3: dick

-1 oder auslassen des Parameters: ändert diese Rahmenstärke nicht.

Block: bestimmt den Block, für den die Formatierung geändert werden soll. Fehlt diese Angabe, so wird der aktuelle Block angenommen.

Wird kein Parameter angegeben, so entspricht dies <Format-Rahmen> - es wird das entsprechende Fenster zur Auswahl angezeigt. (Weitere Hinweise hierzu siehe bei "Makro-Befehle" in der Einleitung!)

Tip: Möchten Sie nur den Rand eines Blocks ändern, so verwenden Sie dazu die Funktion KASTEN.

Beispiel:

FRAME(;;3;3;A10) Die Zelle A10 erhält einen oberen und unteren Rand

RAHMEN(1;1;1;1;A1:A10) Die Zellen A1 bis A10 erhalten einen dünnen Rand auf allen Seiten.

Verwandte Befehle:

KASTEN , **FARBEN** , **SCHRIFT** , **STILWECHSELN** , **AUSRICHTUNG** , **ZAHLENFORMAT**

1.37 SCHRIFT([Num];[Zeichensatz];[Block])

SCHRIFT([Num];[Zeichensatz];[Block])

(Englisch: FONT)

Bestimmt den Textstil und die Schriftart des aktuellen bzw. mit Block angegebenen Blockes und zwar:

Num: Bestimmt den Stil, in dem die Texte erscheinen sollen. Es sind: 0=Normal, 1=Unterstrichen, 2=Fett, 4=Kursiv - weiterhin können alle Kombinationen erreicht werden, indem die entsprechenden Merkmale addiert werden (also: Fett, Kursiv und Unterstrichen = 7) (-1 oder weglassen des Parameters ändert die Formatierung nicht)

Zeichensatz: Gibt an, welcher Zeichensatz benutzt werden soll. Dies ist ein Text, der aus dem Zeichensatznamen (mit oder ohne ".FONT") und einem Schrägstrich gefolgt von der Größe besteht (also etwa "topaz/8" oder "Helvetica.font/13"). Fehlt dieser Parameter, so wird der aktuelle Zeichensatz nicht geändert - wird "" übergeben, so wird der Standard-Zeichensatz eingestellt.

Block: bestimmt den Block, für den die Formatierung geändert werden soll. Fehlt diese Angabe, so wird der aktuelle Block angenommen.

Wird kein Parameter mit angegeben, so entspricht dies <Format-Schriftstil> - es wird das entsprechende Fenster zur Auswahl angezeigt. (Weitere Hinweise hierzu siehe bei "Makro-Befehle" in der Einleitung!)

Beispiel:

SCHRIFT(3) Formatiert die Texte des aktuellen Blocks Fett und Unterstrichen.

SCHRIFT("Times/13";A1:A10) Die Inhalte der Zellen A1 bis A10 werden im Zeichensatz "Times" der Größe 13 dargestellt - die Stilrichtung wird nicht geändert.

FONT() es erscheint ein Fenster, in dem der Benutzer die Formatierung auswählen kann.

Verwandte Befehle:

FARBEN , RAHMEN , STILWECHSELN , AUSRICHTUNG , ZAHLENFORMAT

1.38 SPALTENBREITE(Breite;[Block])

SPALTENBREITE(Breite;[Block])

(Englisch: COLUMNWIDTH)

Bestimmt die Spaltenbreite der aktuellen bzw. mit Block angegebenen Spalten und zwar:

Breite: Bestimmt die Breite der Spalte in 1/8 Zeichen - ist der Wert zu klein oder zu groß, so wird er ignoriert! (außer 0, dies entspricht der Standardbreite!)

Block: bestimmt den Block, für den die Formatierung geändert werden soll. Fehlt diese Angabe, so wird der aktuelle Block angenommen.

Wird kein Parameter angegeben, so entspricht dies <Format-Spaltenbreite> - es wird das entsprechende Fenster zur Auswahl angezeigt. (Weitere Hinweise hierzu siehe bei "Makro-Befehle" in der Einleitung!)

Beispiel:

SPALTENBREITE(80) Die aktuellen Spalte (d.h. alle teilweise markierten Spalten) werden auf eine Breite von 10 Zeichen gesetzt

SPALTENBREITE(0;A1) Spalte A bekommt die Standard-Breite.

SPALTENBREITE() es erscheint ein Fenster, in dem der Benutzer die Formatierung auswählen kann.

Verwandte Befehle:

ZEILENHÖHE

1.39 STDSCHRIFT(Zeichensatz)

STDSCHRIFT(Zeichensatz)

(Englisch: STDFONT)

Hiermit können Sie den Standardzeichensatz für die jeweilige Tabelle festlegen.

Zeichensatz: Ist ein Zeichensatztext (Name/Größe, etwa "Helvetica/13"). Wird dieser Parameter nicht angegeben, so erscheint ein Auswahlfenster.

Ohne Parameter entspricht dies dem Menüpunkt <Format-Standardschrift>.

1.40 STILWECHSELN(Num;[Block])

STILWECHSELN(Num;[Block])

(Englisch: CHANGESTYLE)

Ändert den Textstil des aktuellen bzw. mit Block angegebenen Blocks und zwar:

Num: Bestimmt den Stil, der bei den aktuellen Texten geändert werden soll. Es sind: 1=Unterstrichen, 2=Fett, 4=Kursiv - weiterhin können alle Kombinationen erreicht werden, indem die entsprechenden Merkmale addiert werden (also: Fett, Kursiv und Unterstrichen = 7)

Hinweis: Im Gegensatz zu SCHRIFT, der eine Stilart setzt, wird hier die Stilart geändert, d.h. war das Stilmerkmal zuvor vorhanden, so wird es abgeschaltet, ansonsten eingeschaltet.

Block: bestimmt den Block, für den die Formatierung geändert werden soll. Fehlt diese Angabe, so wird der aktuelle Block angenommen.

Beispiel:

STILWECHSELN(3) Wechselt die Stilmerkmale Fett und Unterstrichen des aktuellen Blocks. (Bei einem normalen Text werden sie gesetzt).

CHANGESTYLE(5) und damit Unterstrichen wieder aus- und Kursiv eingeschalten (Also: Fett und Kursiv)

Verwandte Befehle:

SCHRIFT , **AUSRICHTUNG** , **ZAHLENFORMAT**

1.41 VERBERGEN(Zeile;[Block])

VERBERGEN(Zeile;[Block])

(Englisch: HIDE)

Verbirgt die aktuelle Spalte bzw. Zeile:

Zeile: Bestimmt, ob die Zeile bzw. die Spalte nicht mehr angezeigt werden soll:

0: Spalte

1: Zeile

Block: Bestimmt den Block, auf den der Befehl angewendet werden soll (Je nach Wert von Zeile wird nur die Zeile bzw. Spalte verwandt). Fehlt diese Angabe, so wird der aktuelle Block bzw. die aktuelle Zelle benutzt.

Dies entspricht den Menüpunkten <Format-Verbergen-Spalte> bzw. <Format-Verbergen-Zeile>.

Beispiel:

VERBERGEN(0) blendet die aktuelle Spalte aus.

HIDE(1;C2:C3) blendet die Zeilen 2 und 3 aus.

1.42 ZAHLENFORMAT(Format;[Block])

ZAHLENFORMAT(Format;[Block])

(Englisch: NUMERICFORMAT)

Bestimmt das Zahlenformat des aktuellen bzw. mit Block angegebenen Blocks und zwar:

Zahlenformat: 0=Standard, 1 bis 56 ist die Nummer des gewünschten Zahlenformats (ab 0=Standard durchnummeriert):

Hinweis: Ab TurboCalc3.5 sind einige neuen Formate hinzugekommen (41 bis 56). Aus Kompatibilitätsgründen wurden diese nicht nach der Art "einsortiert", sondern am Ende angefügt.

0 Standard

1 0

2 0,0

...

7 0,000000

8 0,000

9 0.000,00

10 000

11 0000

12 00000

13 0,0E+00

...

18 0,000000E+00

19 0%

...

25 0,000000%

26 0 DM

27 0,00 DM

28 0,000 DM

29 0.000,00 DM

30 TT.MM.JJJJ

31 TT.MM.JJ

32 TT. MMM JJJJ

33 TT. MMM JJ

34 TT. MMM

35 MMM.JJJJ

36 MMM. JJ

37 hh:mm:ss

38 hh:mm

39 h:mm:ss

40 h:mm

Die neuen Formate (ab TurboCalc 3.5):

41 000000

42 0000000

43 00000000

44 000000000

45 0000000000

46 TT M JJJ

47 TT M JJ

48 M JJJ

49 M JJ

50 TT M

51 h:mm:ss am/pm

52 h:mm am/pm

53 [h]

54 [h]:mm

55 [m]

56 [m]:ss

Block: bestimmt den Block, für den die Formatierung geändert werden soll. Fehlt diese Angabe, so wird der aktuelle Block angenommen.

Wird kein Parameter angegeben, so entspricht dies <Format-Zahlenformat> - es wird das entsprechende Fenster zur Auswahl angezeigt. (Weitere Hinweise hierzu siehe bei "Makro-Befehle" in der Einleitung!)

Beispiel:

ZAHLENFORMAT(0) Die aktuellen Zellen werden als Standard dargestellt

NUMERICFORMAT(2;A1:A10) Format "0.00" für die Zellen A1:A10

ZAHLENFORMAT() es erscheint ein Fenster, in dem der Benutzer die Formatierung auswählen kann.

Verwandte Befehle:

SCHRIFT , **AUSRICHTUNG** , **FARBEN** , **RAHMEN** , **KASTEN**

1.43 ZEILENHÖHE(Höhe;[Block])

ZEILENHÖHE(Höhe;[Block])

(Englisch: ROWHEIGHT)

Bestimmt die Zeilenhöhe der aktuellen bzw. mit Block angegebenen Zeilen und zwar:

Höhe: Bestimmt die Höhe der Zeile in Zeichensatzhöhe, es können auch Kommazahlen eingegeben werden! Ist der Wert zu klein oder zu groß, so wird er ignoriert! (Außer 0, dies entspricht der Standardhöhe!)

Block: bestimmt den Block, für den die Formatierung geändert werden soll. Fehlt diese Angabe, so wird der aktuelle Block angenommen.

Wird kein Parameter angegeben, so entspricht dies <Format-Zeilenhöhe> - es wird das entsprechende Fenster zur Auswahl angezeigt. (Weitere Hinweise hierzu siehe bei "Makro-Befehle" in der Einleitung!)

Beispiel:

ZEILENHÖHE(4) Die aktuellen Zeilen (d.h. alle teilweise markierten Zeilen) werden auf eine Höhe von 4 mal Zeichensatzhöhe gesetzt

ZEILENHÖHE(1,5;A1) Zeile 1 bekommt eine Höhe von 1,5 Zeilen.

ZEILENHÖHE() es erscheint ein Fenster, in dem der Benutzer die Formatierung auswählen kann.

Verwandte Befehle:

SPALTENBREITE

1.44 ZELLSCHUTZ([Schreiben];[Formel];[Block])

ZELLSCHUTZ([Schreiben];[Formel];[Block])

(Englisch: PROTECTION)

Bestimmt den Zellschutz des aktuellen bzw. mit Block angegebenen Blocks und zwar:

Schreiben: Bestimmt, ob auf diese Zellen ein "Schreibzugriff" möglich ist. (0=Ja, 1=Nein, -1=nicht ändern.)

Formel: Bestimmt, ob bei diesen Zellen die Formeln verborgen werden (0=ja, 1=Nein, -1=nicht ändern.)

Näheres hierzu siehe bei <Format-Zellschutz>

Block: bestimmt den Block, für den die Formatierung geändert werden soll. Fehlt diese Angabe, so wird der aktuelle Block angenommen.

Wird kein Parameter mit angegeben, so entspricht dies <Format-Zellschutz> - es wird das entsprechende Fenster zur Auswahl angezeigt. (Weitere Hinweise hierzu siehe bei "Makro-Befehle" in der Einleitung!)

Beispiel:

ZELLSCHUTZ(0) Die aktuellen Zellen können, falls die Tabelle geschützt wird, trotzdem geändert werden.

PROTECTION(;1;A1:A10) Es werden keine Formeln der Zellen A1 bis A10 angezeigt!

ZELLSCHUTZ() es erscheint ein Fenster, in dem der Benutzer die Formatierung auswählen kann.

Verwandte Befehle:

SCHRIFT , AUSRICHTUNG , ZAHLENFORMAT , FARBEN , RAHMEN , KASTEN

1.45 Cursorsteuerung

Cursorsteuerung

AKTIVEZELLE()

AUSWÄHLEN([Block])

INSPALTE(Spalte)

INZEILE(Zeile)

LINKS(Num)

SPALTE(Spalte)

LETZTESPALTE()

SUCHEN(Text;Teil;Groß;Spalten;Bereich)

ZEILE(Zeile)

LETZTEZEILE()

1.46 AKTIVEZELLE()

AKTIVEZELLE()

(Englisch: CURRENTCELL)

Verschiebt den sichtbaren Ausschnitt des Bildschirms, so daß die aktuelle Zelle bzw. der aktuelle Block sichtbar wird. (Ist er schon sichtbar, so hat dieser Befehl keine Wirkung)

Dies entspricht dem Menüpunkt <Befehle-Aktive Zelle zeigen>.

Verwandte Befehle:

AUSWÄHLEN

1.47 AUSWÄHLEN([Block])

AUSWÄHLEN([Block])

(Englisch: SELECT oder SELECTCELL oder SELECTRANGE)

Positioniert den Zell-Cursor an die mit Block angegebene Stelle (falls es sich um eine Zelle handelt) bzw. markiert den Block und positioniert den Zell-Cursor in die linke obere Ecke des Blocks. Ist dieser Bereich nicht sichtbar, so wird der Bildschirm automatisch gescrollt.

Block: Bestimmt die neue Cursor-Position. Kann Zellverweis oder Block sein (wie überall natürlich auch ein Name, etwa DATENBANK)

Hinweis: Für ARexx wurden noch die Synonyme SELECTCELL oder SELECTRANGE eingeführt, da dort SELECT ein reserviertes Wort ist und sonst immer in Anführungszeichen gesetzt werden sollte. Es empfiehlt sich also, dort SELECTCELL zu benutzen.

Wird kein Parameter angegeben, so entspricht dies <Befehle-Gehezu> - es wird das entsprechende Fenster zur Auswahl angezeigt. (Weitere Hinweise hierzu siehe bei "Makro-Befehle" in der Einleitung!)

Verwandte Befehle:

AKTIVEZELLE

1.48 INSPALTE(Spalte)

INSPALTE(Spalte)

(Englisch: GOTOCOLUMN)

Setzt den Cursor der aktuellen Tabelle in die Spalte Spalte (1 ist dabei die Spalte ganz links).

Verwandte Befehle:

INZEILE , SPALTE , ZEILE , HOCH , RUNTER , LINKS , RECHTS , AUSWÄHLEN

1.49 INZEILE(Zeile)

INZEILE(Zeile)

(Englisch: GOTOLINE)

Setzt den Cursor der aktuellen Tabelle in die Zeile Zeile (1 ist dabei die oberste Zeile).

Verwandte Befehle:

INSPALTE , SPALTE , ZEILE , HOCH , RUNTER , LINKS , RECHTS , AUSWÄHLEN

1.50 LINKS(Num)

LINKS(Num)

(Englisch: LEFT)

Bewegt den Cursor um Num in die entsprechende Richtung. Fehlt Num, so wird der Cursor um ein Feld bewegt.

Num: Kann eine beliebige (positive) Zahl sein. Fehlt sie, so wird 1 angenommen.

Achtung: LINKS und RECHTS (bzw. LEFT und RIGHT) können als Makrobefehl nicht benutzt werden (nur als ARexxbefehl!), da entsprechende Funktionen gleichen Namens (aber mit anderer Funktion und anderen Parametern) existieren. Benutzen Sie hierzu SPALTE(-1) bzw. SPALTE(1)

Verwandte Befehle:

ZEILE , SPALTE , INZEILE , INSPALTE , AUSWÄHLEN

1.51 RECHTS(Num)

RECHTS(Num)

(Englisch: RIGHT)

Bewegt den Cursor um Num in die entsprechende Richtung. Fehlt Num, so wird der Cursor um ein Feld bewegt.

Num: Kann eine beliebige (positive) Zahl sein. Fehlt sie, so wird 1 angenommen.

Achtung: LINKS und RECHTS (bzw. LEFT und RIGHT) können als Makrobefehl nicht benutzt werden (nur als ARexxbefehl!), da entsprechende Funktionen gleichen Namens (aber mit anderer Funktion und anderen Parametern) existieren. Benutzen Sie hierzu SPALTE(-1) bzw. SPALTE(1)

Verwandte Befehle:

ZEILE , SPALTE , INZEILE , INSPALTE , AUSWÄHLEN

1.52 HOCH(Num)

HOCH(Num)

(Englisch: UP)

Bewegt den Cursor um Num in die entsprechende Richtung. Fehlt Num, so wird der Cursor um ein Feld bewegt.

Num: Kann eine beliebige (positive) Zahl sein. Fehlt sie, so wird 1 angenommen.

Verwandte Befehle:

ZEILE , SPALTE , INZEILE , INSPALTE , AUSWÄHLEN

1.53 RUNTER(Num)

RUNTER(Num)

(Englisch: DOWN)

Bewegt den Cursor um Num in die entsprechende Richtung. Fehlt Num, so wird der Cursor um ein Feld bewegt.

Num: Kann eine beliebige (positive) Zahl sein. Fehlt sie, so wird 1 angenommen.

Verwandte Befehle:

ZEILE , SPALTE , INZEILE , INSPALTE , AUSWÄHLEN

1.54 SPALTE(Spalte)

SPALTE(Spalte)

(Englisch: COLUMN)

Setzt den Cursor der aktuellen Tabelle Spalte Spalten links bzw. rechts (1 bedeutet eine Spalte nach rechts).

Verwandte Befehle:

ZEILE , **INZEILE** , **INSPALTE** , **HOCH** , **RUNTER** , **LINKS** , **RECHTS** , **AUSWÄHLEN**

1.55 LETZTESPALTE()

LETZTESPALTE()

(Englisch: LASTCOLUMN)

Positioniert den Cursor, ausgehend von seiner aktuellen Position, in die letzte Spalte dieser Zeile, in der sich noch Daten befinden.

Verwandte Befehle:

ZEILE , **INZEILE** , **LETZTEZEILE** , **INSPALTE** , **HOCH** , **RUNTER** , **LINKS** , **RECHTS** , **AUSWÄHLEN**

1.56 SUCHEN(Text;Teil;Groß;Spalten;Bereich)

SUCHEN(Text;Teil;Groß;Spalten;Bereich)

(Englisch: FIND)

Sucht nach dem nächsten Auftreten von Text und setzt den Cursor in diese Zelle.

Text: Ist der Text, nach dem gesucht werden soll. Wird dieser Parameter ausgelassen, so wird der zuvor eingegebene Text verwandt, war noch kein Text vorhanden, so erscheint eine Fehlermeldung. (siehe dazu auch drittes Beispiel unten)

Teil: Bestimmt, ob die Zelle genau den Text enthalten muß (0 oder FALSCH), oder ob es reicht, wenn ein Teil der Zelle der Text Text ist (WAHR bzw. ungleich Null). (Falls dieser Parameter ausgelassen wird, wird WAHR angenommen.)

Groß: Bestimmt, ob die Groß- und Kleinschreibung übereinstimmen muß (0 oder FALSCH), oder ob es reicht, wenn der Text (ohne Groß/Kleinschreibung) stimmt (WAHR bzw. ungleich Null). (Falls dieser Parameter ausgelassen wird, wird WAHR angenommen.)

Spalten: Bestimmt, ob spaltenweise (0 oder FALSCH) oder zeilenweise (WAHR bzw. ungleich Null) nach dem Text gesucht werden soll. (Falls dieser Parameter ausgelassen wird, wird zeilenweise angenommen.)

Bereich: Wird eine Zelle angegeben, so bestimmt diese Angabe, ab welcher Zelle mit der Suche begonnen werden soll.

Falls ein Bereich angegeben wird, so wird die Suche in der linken oberen Ecke des Bereichs gestartet und es wird nur innerhalb dieses Bereichs gesucht.

Ohne Parameter entspricht dieser Befehl dem Menüpunkt <Befehle-Suchen...>

Im Makromodus: Wird dieser Befehl im Makromodus benutzt, so wird der Zelle, in der dieser Befehl steht, der Wert WAHR zugewiesen, falls die Suche erfolgreich war. Wurde nichts gefunden, so enthält diese Zelle den Wert FALSCH. Dies kann dann mit SPRINGEWENN abgefragt werden (siehe bei "Rückgabewerte von Befehlen" am Beginn der Befehlsübersicht).

Beispiel:

SUCHEN("hallo")

sucht nach dem Text "hallo", die Großschreibung ist dabei beliebig und der Text kann auch als Teil eines Textes vorkommen (etwa "Hallo Alex").

SUCHEN("hallo";;0)

dito, jetzt muß das Wort jedoch klein geschrieben sein, um gefunden zu werden ("hallo Alex").

SUCHEN(;))

setzt die letzte Suche fort.

1.57 ZEILE(Zeile)

ZEILE(Zeile)

(Englisch: LINE)

Setzt den Cursor der aktuellen Tabelle Zeile Zeilen hoch bzw. runter (1 bedeutet eine Zeile nach unten).

Verwandte Befehle:

SPALTE , INZEILE , INSPALTE , HOCH , RUNTER , LINKS , RECHTS , AUSWÄHLEN

1.58 LETZTEZEILE()

LETZTEZEILE()

(Englisch: LASTROW)

Setzt den Cursor, ausgehend von seiner aktuellen Position, in die letzte Zeile der aktuellen Spalte, in der sich noch Daten befinden.

Verwandte Befehle:

SPALTE , LETZTESPALTE , ZEILE , INZEILE , HOCH , RUNTER , LINKS , RECHTS , AUSWÄHLEN

1.59 Eingabe

Eingabe

AMIGAGUIDE(Datei;Kommando)

ANFRAGE(Text[;Titel])

DIALOG(Dialog;Hook)

DLGWEITER(Flag)

EINGABE(Text[;Titel];[Zelle])

HILFE(Num;Datei)

MELDUNG(Text[;Titel])

PAUSE(Zeit)

REQPARA(X;Y;Oktext;Abbruchtext;Zeit)

SCHREIBE(Inhalt[;Zelle])

SIGNAL()

1.60 AMIGAGUIDE(Datei;Kommando)

AMIGAGUIDE(Datei;Kommando)

Damit können (bestimmte Teile einer) AmigaGuide-Dateien angezeigt werden. Dies wird etwa dazu benutzt, die AmigaGuide-Hilfefunktion von TurboCalc anzuzeigen.

Datei: Ist der Name der AmigaGuide-Datei. Entweder kompletten Pfad angeben oder relativ zu dem Verzeichnis, in dem sich TurboCalc befindet. Wird dieser Parameter ausgelassen, so wird "TurboCalc.guide" benutzt.

Kommando: Ist ein Kommando, das an den AmigaGuide-Betrachter geschickt werden soll. Mögliche Kommandos sind:

ALINK <name> Lädt den Abschnitt <name> in ein neues Fenster. Anmerkung: In der Version 39 der 'amigaguide.library' wird kein neues Fenster geöffnet (dies ist ein dokumentierter Fehler dieser Version).

LINK <name> Lädt den Abschnitt <name>.

RX <macro> Führt ein ARexx-Makro aus.

RXS <cmd> Führt einen ARexx-String aus, etwa:

'ADDRESS COMMAND DISPLAY <picture name>'

'ADDRESS COMMAND MORE <doc>'

CLOSE Schließt das Fenster (möglichst nur für Fenster, die mit ALINK geöffnet wurden, benutzen)

QUIT Schließt die aktuelle AmigaGuide-Datei.

Hauptsächlich dürfte wohl 'LINK <name>' benutzt werden, um einen bestimmten Abschnitt der AmigaGuide-Datei anzuzeigen. (<name> ist dabei der Teil, der sich bei @ENDNODE

1.61 befindet).

Das Anzeigen findet dabei im Hintergrund (asynchron) statt. Es kann normal mit TurboCalc weitergearbeitet werden. Auch weitere AMIGAGUIDE-Befehle sind möglich. Wird dabei der gleiche Dateinamen angegeben, so wird kein neuer AmigaGuide-Betrachter geladen, sondern nur den angegebenen Befehl an den alten Betrachter geschickt. Wird ein Dateiname benutzt, der noch nicht zuvor benutzt wurde, so wird ein neuer Betrachter (d.h. ein neues Fenster) gestartet.

Anmerkung: Zum Anzeigen wird die 'amigaguide.library' benötigt. Ansonsten erscheint eine entsprechende Fehlermeldung.

Beispiel:

```
=AMIGAGUIDE("TurboCalc.guide")
```

Öffnet ein Fenster mit dem "Inhalt"

Inhalt

(main) von TurboCalc.guide

```
=AMIGAGUIDE("TurboCalc.guide";"LINK save")
```

Zeigt in dem zuvor geöffneten Fenster den Inhalt des Abschnitts (node) "save" an.

```
=AMIGAGUIDE("Test.guide")
```

Öffnet eine weitere AmigaGuide-Datei und zeigt "main" an.

1.62 ANFRAGE(Text[;Titel])

ANFRAGE(Text[;Titel])

(Englisch: REQUEST)

Öffnet ein Fenster und zeigt den Text Text in diesem Fenster an. Mit Titel kann der Fenstertitel bestimmt werden. Das Fenster erhält die Gadgets >OK< und >Abbruch<.

Text: Ist ein normaler Text (in Anführungszeichen bzw. als Text-Formel). Dieser wird in dem Fenster gedruckt - ist er länger als eine Zeile, so wird er automatisch getrennt. (Die Tilde "~" kann als Hilfs-Trennstrich verwandt werden: Sie wird nicht gedruckt, außer am Zeilenende, dann wird daraus ein Minus "-".Das "umgedrehte P" (ASCII-Code 182, zu erzeugen über <Alt>+<P>) bedeutet "Zeilenende", danach wird der Text in der nächsten Zeile fortgesetzt).

Titel: Bestimmt den Fenstertitel. Kann auch weggelassen werden, dann erscheint ein Standardtitel.

Tip: Möchte man Zellinhalte in den Text einbeziehen, so kann man dazu die Funktion TEXT(Zelle) und die Konkatenation durch "+" benutzen, etwa "Inhalt der Zelle A1 ist"+TEXT(A1)+"!". Siehe dazu auch drittes Beispiel bei MELDUNG.

Tip: Zum Ändern der Fensterposition etc siehe REQPARA.

Im Makromodus: Wird dieser Befehl im Makromodus benutzt, so wird der Zelle, in der dieser Befehl steht, der Wert WAHR zugewiesen, falls auf >OK< geklickt wurde. Wurde >Abbruch< angeklickt bzw. das Fenster geschlossen, so enthält diese Zelle den Wert FALSCH. Dies kann dann mit SPRINGEWENN abgefragt werden (siehe Beispiel unten).

Im ARexx-Modus: Bei >OK< wird keine Fehlermeldung erzeugt, ansonsten der Fehler ABORT zurückgegeben.

Beispiel:

```
REQUEST("Dies ist ein Test";"Titel")
```

Öffnet ein Fenster mit dem Titel und zeigt "Dies ist ein Test" an. Es kann >OK< oder >Abbruch< ausgewählt werden.

Folgender Programmabschnitt realisiert eine Sicherheitsabfrage:

(Beachten Sie dabei bitte das "#A10" zur Abfrage der Zelle A10 im Makroblatt (und nicht in der aktiven Zelle)

```
A10 =REQUEST("Wollen Sie dies wirklich tun?";"Testmakro")
```

```
A11 =SPRINGEWENN(NOT(#A10);A20)
```

... hier kommt das eigentliche

... Programm...

```
A20 =RETURN()
```

1.63 DIALOG(Dialog;Hook)

DIALOG(Dialog;Hook)

(English: DIALOGUE)

Damit kann (innerhalb eines Makros) ein Benutzerdialog angezeigt werden, der es erlaubt, komplexere Abfragen (als etwa EINGABE) durchzuführen.

Dialog ist ein Zellverweis auf die linke obere Ecke eines Bereichs, der die Dialogbeschreibung enthält (siehe unten)

Hook: Bestimmt eine Routine, die bei Änderung des Requester aufgerufen wird (etwa falls ein Boolean-Gadget angeklickt wird). Der Requester bleibt in diesem Fall geöffnet. (Ein Anwendungsbeispiel ist etwa das AutoFormat-Makro, in dem nach Auswahl der entsprechenden Formatierung diese sofort ausgeführt wird, ohne den Requester schließen zu müssen)

Vor Aufruf der Hook-Routine werden die Dialogfelder mit den Ergebnissen aktualisiert.

Die Hook-Routine muß mit DLGWEITER (DLGRESUME) abgeschlossen werden, um eine weitere Abfrage zu ermöglichen. Innerhalb dieser Hook-Schleife darf kein weiterer Dialog aufgerufen werden!

Hinweis: Die Hook-Routine sollte recht kurz sowie fehlerfrei sein!

Siehe dazu auch das Beispielmakro 'Makro_Dialog2' sowie 'Makro_AutoFormat'.

Es öffnet sich ein Fenster, in dem sich die mit Dialog angegebenen Elemente (etwa Texte, Gadgets, etc) befinden. Das Makro wird erst nach Klick auf OK bzw. eines anderen Gadgets fortgesetzt.

Wird dieser Befehl innerhalb eines Makros aufgerufen, so enthält die Zelle, in der der Makrobefehl steht, den Rückgabewert (siehe etwa ANFRAGE).

Klick auf OK (bzw. <Return>) ergibt 0,

Klick auf Abbruch (bzw. <ESC>) ergibt -1,

Klick auf ein anderes Gadget (GAD-Feld) ergibt den jeweiligen Parameter, siehe unten!

Dialogbereich:

Dies ist eine 8 Felder breite Tabelle mit folgenden Einträgen:

```
pos0 pos1 pos2 pos3 pos4 pos5 pos6 pos7
```

```
Befehl X Y Breite Höhe Text Para Daten
```

Nicht immer werden alle Daten einer Zeile benutzt, die unbenutzten Einträge werden ignoriert. Einzelheiten finden Sie bei der Beschreibung der einzelnen Befehle.

X/Y/Breite/Höhe: Bestimmt die Position und die Größe des jeweiligen Elements. Alle Angaben (positiv!) beziehen sich auf 1/8 Zeichensatzbreite bzw. Höhe! (Somit können also Zeichensatzunabhängige Dialoge erstellt werden)

Tip: Arbeitet man mit Topaz/8 (bzw. einem anderen 8-Punkte-Zeichensatz), so entsprechen die Werte genau den Bildschirm-"Pixeln".

Breite/Höhe kann auch ausgelassen werden (bzw. auf 0 gesetzt werden), dann wird die Standardhöhe und -breite genommen. Dies ist oft zu empfehlen!

Für X/Y können auch negative Werte (oder 0) angegeben werden. Dann gibt dies die Änderung zu den vorherigen Werten an. (Anmerkung: Die Änderung ist immer nach links unten! Es können auch

gemischte Werte (pos/neg) für X/Y angegeben werden) (Siehe Beispiel 1)

Achtung: 0 wird als "negativ" interpretiert und bedeutet, "die Position nicht zu ändern!" Will man die "erste Position links oben" direkt angeben, so muß dazu also 1 benutzt werden!

Befehl ist eine Nummer mit dem Befehl:

0 END Dies beendet die Dialogtabelle! Steht an der 'Text'-Stelle dieses Befehls ein Text, so wird dieser als Titel benutzt, Beispiel siehe unten.

1 TEXT Fügt einen Text (von pos5) an die X/Y-Stelle ein.

2 GAD Fügt ein Gadget mit dem Text (von pos5) ein. Para (pos6) gibt den Rückgabewert an, der bei Klick auf dieses Gadget zurückgegeben wird (dadurch wird auch der Dialog geschlossen)

3 BOOL (SCHALTER) Erzeugt einen Schalter mit dem Text (von pos5). Daten (pos7) erhält den Auswahlmodus (0=aus). Dieser wird beim Beenden des Dialogs zurückgeschrieben (falls Rückgabewert nicht negativ)

4 CHOICE (AUSWAHL) Ähnlich BOOL, jedoch kann hiermit zwischen mehreren Möglichkeiten gewählt werden. Dazu wird mit dem Parameter Mask (pos6) angegeben, welche anderen CHOICE-Felder bei Auswahl dieses Feldes automatisch abgewählt werden: Mask berechnet sich als Summe der 2^i , wobei i die Stellen (ab 1 gezählt) der CHOICE-Felder in dieser Tabelle darstellen (Achtung: TEXT wird nicht gezählt!!!)

5 STRING Erzeugt ein Textfeld zur Eingabe eines Textes. Vorgabe- und Rückgabewert in Daten (pos7). Para (pos6) enthält die maximale Breite (in Zeichen) des Textes (maximal 300 Zeichen). Dies kann größer als die Breite des Textfeldes sein, dann wird jeweils gescrollt!

6 MULT In dieser Version noch nicht implementiert!

7 LIST In dieser Version noch nicht implementiert!

8 FRAME (RAHMEN) Es wird ein Rahmen eingefügt (mit Breite/Höhe). Text (pos5) erscheint im oberen Rahmenstrich.

9 SEPLINE (TRENNSTRICH) Es wird an die aktuelle Y-Position ein Strich über die ganze Breite eingefügt.

10 OKABORT (OK_ABBRUCH) Dies fügt einen Trennstrich und zwei Gadget mit den Titeln "OK" und "Abbruch" ein. Dies wurde also sozusagen als "Makro" hinzugefügt, da es relativ oft vorkommt!

Achtung: Parameter-Daten werden nur geändert, falls der Rückgabewert größer als 0 ist (also etwa nicht bei Abbruch!) Weiterhin wird in diesem Fall kein "Neuberechnen" und auch kein "Neudarstellen" dieser Daten durchgeführt. Da dieser Befehl jedoch (fast) immer von einem Makro aus durchgeführt wird, sollte dies nicht ins Gewicht fallen (sondern im Gegenteil eher das Makro beschleunigen!) Ein Abfragen der Daten vom Makro aus ist natürlich ohne Probleme möglich (auch ein "NEUBERECHNEN" von Makro aus).

Beispiel 1 (negative Koordinaten)

A B C D E F

1 8 8 0 0 Dies ist ein Text

1 -32 -16 0 0 ein zweiter Text

2 0 -16 0 0 Gad1

2 -32 0 0 0 Gad2

0

Dies schreibt an die Position 8/8 einen ersten Text und dann darunter (zwei Zeilen) und 4 Zeichen links den zweiten Text. Weitere zwei Zeilen darunter wird das Gadget "Gad1" mit Standardbreite eingefügt und dann mit 4 Zeichen Abstand in die gleiche Zeile das Gadget "Gad2".

Titel:

A B C D E F

... (andere Befehle)

0 Beispieltitel

Vergleiche hierzu auch Makro_Dialog sowie Makro_Dialog2.

1.64 DLGWEITER(Flag)

DLGWEITER(Flag)

(Englisch: DLGRESUME)

Dieses beenden die Hook-Routine des DIALOG-Befehls (siehe zuvor).

Flag bestimmt, ob der Dialog geschlossen werden soll:

0: Der Dialog wird geschlossen und die Ausführung unmittelbar nach dem DLGWEITER-Befehl fortgesetzt (ggf. von dort zurück zum Befehl nach DIALOG verzweigen)

1: Der Dialog wird normal weitergeführt. Dies ist der Standardwert, der auch bei Auslassen des Parameters angenommen wird.

Hinweis: Ist bei Ausführen des DLGWEITER-Befehls kein Dialog aktiv (d.h. der Befehl befindet sich nicht in der Hook-Routine), so wird er einfach ignoriert, d.h. nichts getan.

1.65 EINGABE(Text[;Titel];[Zelle])

EINGABE(Text[;Titel];[Zelle])

(Englisch: INPUT)

Öffnet ein Fenster und zeigt den Text Text in diesem Fenster an. Mit Titel kann der Fenstertitel bestimmt werden. Das Fenster erhält die Gadgets >OK< und >Abbruch< sowie ein Textfeld, in dem der Benutzer Eingaben vornehmen kann. Diese wird dann (bei >OK< oder >Return<) in die angegebene bzw. aktuelle Zelle geschrieben.

Text: Ist ein normaler Text (in Anführungszeichen bzw. als Text-Formel). Dieser wird in dem Fenster gedruckt - ist er länger als eine Zeile, so wird er automatisch getrennt. (Die Tilde "~" kann als Hilfs-Trennstrich verwandt werden: Sie wird nicht gedruckt, außer am Zeilenende, dann wird daraus ein Minus "-". Das "umgedrehte P" (ASCII-Code 182, zu erzeugen über <Alt>+<P>) bedeutet "Zeilenende", danach wird der Text in der nächsten Zeile fortgesetzt).

Titel: Bestimmt den Fenstertitel. Kann auch weggelassen werden, dann erscheint ein Standardtitel.

Zelle: Bestimmt, in welche Zelle das Ergebnis bei >OK< übernommen werden soll. Fehlt diese Angabe, so wird die aktuelle Zelle des aktuellen Fensters benutzt. (Es handelt sich dabei normalerweise um das Fenster, aus dem das Makro aufgerufen wurde und nicht um das Makrofenster!)

Tip: Möchte man Zellinhalte in den Text einbeziehen, so kann man dazu die Funktion TEXT(Zelle) und die Konkatenation durch "+" benutzen, etwa "Inhalt der Zelle A1 ist"+TEXT(A1)+"!". Siehe dazu auch drittes Beispiel bei MELDUNG.

Tip: Zum Ändern der Fensterposition etc siehe REQPARA.

Hinweis: Geben Sie diesen Befehl als ARexx-Befehl ein und enthalten der Text oder die Zelle Leerzeichen, so achten Sie darauf, daß sie in Anführungszeichen gesetzt werden. Dazu entweder dann den ganzen Befehl noch in einfache Anführungszeichen setzen, oder aber den Text in einfache und doppelte Anführungszeichen setzen: "'Dies ist ein Text"'. Siehe dazu Beispiel unten.

Im Makromodus: Wird dieser Befehl im Makromodus benutzt, so wird der Zelle, in der dieser Befehl steht, der Wert WAHR zugewiesen, falls auf >OK< geklickt wurde. Wurde >Abbruch< angeklickt bzw. das Fenster geschlossen, so enthält diese Zelle den Wert FALSCH. Dies kann dann mit SPRINGEWENN abgefragt werden (siehe Beispiel unten).

Im ARexx-Modus: Bei >OK< wird keine Fehlermeldung erzeugt, ansonsten der Fehler ABORT zurückgegeben.

In beiden Fällen enthält die aktuelle Zelle die Eingabe. Diese wird dabei nach den normalen Eingaberegeln (siehe "Die Eingabe") bearbeitet. Es können also Zahlen, Datum, Uhrzeit, Wahrheitswerte und Texte (auch >"123< um Zahlen als Text einzugeben) eingegeben werden.

Beispiel:

```
EINGABE("Name eingeben:","Ihr Name";A1)
```

Öffnet ein Fenster mit dem Titel "Ihr Name" und zeigt "Name eingeben:" an. Der Benutzer kann nun seinen Namen eingeben. Bei >OK< wird er in das Feld A1 geschrieben, bei >Abbruch< verworfen.

Folgender Programmausschnitt realisiert eine Namensabfrage:

Es wird nach dem Namen gefragt und falls der Benutzer auf >Abbruch< klickt bzw. keinen korrekten Namen (d.h. keinen Text) eingegeben hat, wird die Routine übersprungen. Ansonsten wird zuerst eine Meldung mit "Hallo Name" angezeigt.

(Beachten Sie dabei bitte das "#A10" zur Abfrage der Zelle A10 im Makroblatt (und nicht in der aktiven Zelle)

```
A10 =INPUT("Ihr Name:","Name eingeben";A1)
```

```
A11 =SPRINGEWENN(NOT(#A10 AND ISSTRING(A1));A20)
```

```
A12 =MELDUNG("Hallo "+A1)
```

```
... Programm...
```

```
A20 =RETURN()
```

```
/* ARexx-Beispiel: */
```

```
Options Results
```

```
ADDRESS TCALC
```

```
EINGABE "Bitte Namen eingeben" "Ihr Name" C5
```

```
GETVALUE C5
```

```
SAY "Hallo" Result
```

Dieses Beispiel fragt nach dem Namen, schreibt ihn nach C5 und druckt dann Hallo Name aus.

(Beachten Sie bitte die einfachen und doppelten Anführungszeichen, um die Texte "Bitte Namen eingeben" und "Ihr Name", da diese Leerzeichen enthalten, siehe auch Hinweis zuvor).

Statt dieser Zeile wäre auch folgende möglich (alles von Hochkommata umschlossen):

```
'EINGABE "Bitte Namen eingeben" "Ihr Name" C5'
```

1.66 HILFE(Num;Datei)

```
HILFE(Num;Datei)
```

(Englisch: HELP)

Dies öffnet eine Textdatei (normalerweise eine Hilfe-Datei, jedoch nicht zwingend nötig) und zeigt sie an:

Num: Ist die Nummer des Hilfetextes in der Datei (siehe weiter unten)

Datei: Ist die Datei, die geöffnet werden soll. Wird dieser Parameter ausgelassen, so wird "TurboCalc.HELP" benutzt.

Die Daten werden dabei nacheinander im aktuellen Verzeichnis, im TurboCalc-Verzeichnis (also in dem sich TurboCalc befindet) und in "S:" gesucht.

Dateiaufbau: In der Hilfedatei handelt es sich um einen reinen "ASCII"-Text, in dem die Zeilen durch CR (d.h. <Return>, Code 10) abgeschlossen werden. Die Zeilen sollten nicht länger als 60 Zeichen sein, der Rest wird abgeschnitten.

Um unterschiedliche "Texte" in einer Datei zusammenfassen zu können (und so nicht unnötig viele Dateien erzeugen zu müssen), wurde das Doppelkreuz ("#") als Trennzeichen eingeführt. Es muß als erstes Zeichen in der Zeile nach einem CR auftreten. Diese "Texte" können dann per Num aufgerufen werden (0 ist der Text bis zum ersten "#", danach kommt 1, usw.)

Tip: Dieser Befehl kann somit auch dazu benutzt werden, um beliebige Textdateien anzuzeigen, siehe zweites Beispiel.

Beispiel:

HILFE(0) zeigt den ersten Hilfetext an.

HILFE(0;"df0:Text")

zeigt die Datei "df0:Text" an (bis zum ersten # am Beginn einer Zeile, doch dies kommt in normalen Texten praktisch nie vor!)

1.67 MELDUNG(Text[;Titel])

MELDUNG(Text[;Titel])

(Englisch: MESSAGE)

Öffnet ein Fenster und zeigt den Text Text in diesem Fenster an. Mit Titel kann der Fenstertitel bestimmt werden. Das Fenster erhält das Gadget >Weiter<. Ein Klick darauf bzw. das Schließen des Fensters setzt das Makro fort.

Text: Ist ein normaler Text (in Anführungszeichen bzw. als Text-Formel). Dieser wird in dem Fenster gedruckt - ist er länger als eine Zeile, so wird er automatisch getrennt. (Die Tilde "~" kann als Hilfs-Trennstrich verwandt werden: Sie wird nicht gedruckt, außer am Zeilenende, dann wird daraus ein Minus "-". Das "umgedrehte P" (ASCII-Code 182, zu erzeugen über <Alt>+<P>) bedeutet "Zeilenende", danach wird der Text in der nächsten Zeile fortgesetzt).

Titel: Bestimmt den Fenstertitel. Kann auch weggelassen werden, dann erscheint ein Standardtitel.

Tip: Möchte man Zellinhalte in den Text einbeziehen, so kann man dazu die Funktion TEXT(Zelle) und die Konkatenation durch "+" benutzen, etwa "Inhalt der Zelle A1 ist"+TEXT(A1)+"!". Siehe dazu auch drittes Beispiel.

Tip: Zum Ändern der Fensterposition etc siehe REQPARA.

Beispiel:

MELDUNG("Dies ist ein Test";"Titel")

Öffnet ein Fenster mit dem Titel und zeigt "Dies ist ein Test" an.

MELDUNG("LangesWortmit~Trennhilfe").

Die Tilde bestimmt, an welcher Stelle der Text getrennt werden kann.

MELDUNG("Zelle A1: "+TEXT(A1)+ZEICHEN(182)+" Zelle A2: "+TEXT(A2)+ZEICHEN(182)+" A1+A2: "+TEXT(A1+A2))

ZEICHEN(182) ergibt dabei das "umgedrehte P" (kann auch direkt über <Alt>-<P> eingegeben werden) und gibt an, daß der Text in einer neuen Zeile fortgesetzt wird.

1.68 PAUSE(Zeit)

PAUSE(Zeit)

(Englisch: DELAY)

Stoppt die Makroausführung für eine gewisse Zeit, etwa um dem Benutzer einen Blick auf die aktuelle Tabelle zu gewähren oder um nach dem Starten von Dosbefehlen (per RUN) kurz zu warten

Zeit: gibt dabei die Zeit in etwa 1/50 Sekunden an, die gewartet werden soll. Das Maximum ist 30*50=1500 (für 30 Sekunden). Dies wurde aus Sicherheitsgründen eingeführt, um TurboCalc nicht zu lang zu blockieren! Ggf. mehrere Delay-Befehle nacheinander aufführen, falls länger gewartet werden soll. Wird keine Zeit angegeben, so wird eine Sekunde gewartet.

Achtung: Das Warten von TurboCalc kann nur im Makromodus unterbrochen werden (dort durch Anwahl von <Makro-Makro stoppen>, dies wird alle 2 Sekunden geprüft). Wird dieser Befehl per ARexx oder aber Direkteingabe (F10) gestartet, so muß die entsprechende Zeit gewartet werden (deswegen auch auf 30 Sekunden beschränkt)

1.69 REQPARA(X;Y;Oktext;Abbruchtext;Zeit)

REQPARA(X;Y;Oktext;Abbruchtext;Zeit)

Hiermit kann die Position, die Gadgets, sowie die angezeigte Zeit des nächsten Requesters (mit MELDUNG, ANFRAGE, EINGABE) bestimmt werden.

Dazu muß dieser Befehl vor dem jeweiligen Requester-Befehl ausgeführt werden.

X, Y: Dies gibt die X bzw. die Y-Koordinate des Fensters an. Positive Werte ergeben "Pixels" ausgehend von der linken oberen Ecke des Fensters, -1 zentriert das Fenster in der jeweiligen Richtung (dies wird auch benutzt, falls der Parameter ausgelassen wird)

Oktext: Dies ist der Text, der im Gadget "OK" (links unten) bzw. bei Meldungen im Gadget "WEITER" (mitte) erscheinen soll. Wird dieser Text ausgelassen bzw. "" angegeben, so wird der Standardtext verwandt.

Abbruchtext: dito, jedoch für den anderen Text ("ABBRUCH"). Bei "MELDUNG" wird dieser ignoriert.

Hinweis: Das Gadget hat immer die gleiche Breite, also bitte den Text entsprechend anpassen.

Zeit: Hiermit wird die Zeit bestimmt, nach der das Fenster automatisch geschlossen werden soll. Die Zahl entspricht in etwa der Zeit in Sekunden, 0 bzw. Auslassen des Parameters entspricht dem Standard (d.h. kein automatisches Schließen) Beim automatischen Schließen wird ein Klick auf "Abbruch" simuliert!

Hinweis: Die Zeit wird nur gezählt, solange das Fenster aktiv ist. Weiterhin ist die Zeit nicht genau, sondern nur ein ungefährender Richtwert. Diese Einschränkungen sollten jedoch für den Zweck der Funktion (eine Meldung nur kurz anzuzeigen) nicht hinderlich sein.

Achtung: REQPARA bezieht sich stets auf das nächste Meldung/Requester-Fenster. Wird dabei statt per MELDUNG, ANFRAGE... etwa eine Fehlermeldung angezeigt, so werden auch hierfür die aktuelle Einstellung benutzt. Dies dürfte jedoch zu keinen größeren Problemen führen.

Beispiel:

```
=REQPARA(;;"Toll";;20)
```

```
=MELDUNG("Hallo")
```

Es erscheint die Meldung "Hallo" - das Fenster wird bei Klick auf "Toll" (so wurde das "Weiter"-Gadget umbenannt) bzw. nach 20 Sekunden automatisch geschlossen.

```
=REQPARA(0;0)
```

```
=MELDUNG("Hallo")
```

Es erscheint die Meldung "Hallo", das Fenster wird dabei in der linken oberen Ecke des Bildschirms geöffnet

1.70 SCHREIBE(Inhalt[;Zelle])

SCHREIBE(Inhalt[;Zelle])

(Englisch: PUT)

Schreibt Inhalt in die angegebene Zelle (bzw. die aktuelle Zelle). Dies entspricht somit der direkten Eingabe dieses Inhalts.

Inhalt: Der Text, der in die Zelle eingegeben werden soll. (Er wird dabei genauso behandelt, als wenn er direkt per Tastatur eingegeben würde - also können auch Datum, Uhrzeit, Wahrheitswerte und Formeln eingegeben werden). All dies muß jedoch von Anführungszeichen umschlossen werden. (Enthält der Text bzw. die Formel Anführungszeichen, so müssen diese durch zwei Anführungszeichen ersetzt werden (siehe Beispiel unten)

Zahlen, Wahrheitswerte können direkt (ohne Anführungszeichen) als Parameter geschrieben werden - somit bleibt eine Umwandlung in Text erspart. Gleiches gilt auch für Datum und Uhrzeit, beachten Sie jedoch, daß diese nicht direkt in Formeln eingegeben werden können (nur mit WERT, DATWERT...)

Zelle: Bestimmt die Zelle, in die der Inhalt geschrieben werden soll. Fehlt sie, so wird die aktuelle Cursorposition verwandt.

Beispiel:

SCHREIBE("12";A1)

SCHREIBE(12;A1)

Beide machen das Gleiche: Schreiben die Zahl 12 in die Zelle A1

PUT(A1+1;A1)

Erhöht den Inhalt der Zelle A1 um 1 (praktisch im Zusammenhang mit IFGOTO, siehe dort)

PUT("=a1+1";A2)

Schreibt in die Zelle A2 die Formel A1+1

PUT("Er sagte: ""Hallo""";A1)

Schreibt in die Zelle A1 den Text Er sagte: "Hallo". (Mit den zwei Anführungszeichen!!!)

PUT("""12"";A1)

Schreibt in die Zelle A1 den Text 12 (d.h. den Text "12, so daß es nicht als Zahl interpretiert wird).

1.71 SIGNAL()

SIGNAL()

(Englisch: BEEP)

Läßt den Bildschirm kurz aufblincken und erzeugt (ab OS2.0) einen Warnton (beides kann per "Sound" bei den WorkBench-Prefs ein- bzw. ausgeschaltet werden).

Dies ist nützlich bei der Makroausführung, falls der Anwender etwas falsch gemacht hat.

1.72 Laden/Speichern

Laden/Speichern

ABLAGENLESEN([Block])

ABLAGESCHREIBEN([Block])

BLOCKSPEICHERN([Block];[Name])

CSVBLOCKSPEICHERN([Block];[Name];[Trennzeichen])

CSVEINFÜGEN([Block];[Name];[Trennzeichen])

CSVLADEN([Name];[Trennzeichen])

CSVSPEICHERN([Name];[Trennzeichen])

KONFIGLADEN()

KONFIGSPEICHERN()

LADEN([Name])

PROCALCEINFÜGEN([Block];[Name])

PROCALCLADEN([Name])

SPEICHERN([Name])

SPEICHERNALS([Name])

SYLKBLOCKSPEICHERN([Block];[Name])

SYLKEINFÜGEN([Block];[Name])

SYLKLADEN([Name])

SYLKSPEICHERN([Name])

TCDEINFÜGEN([Block];[Name])

1.73 ABLAGENLESEN([Block])

ABLAGENLESEN([Block])

(Englisch: CLIPREAD)

Liest aus der Zwischenablage und fügt die Daten ab Block ein. Wird für Block eine Zelle angegeben, so werden die Daten soweit vorhanden nach rechts unten fortgesetzt, ansonsten wird höchstens der mit Block angegebene Bereich gefüllt. Wird Block ausgelassen, so wird der aktuell markierte Bereich verwandt.

Die Einstellungen für den Zwischenablagezugriff können mit ZWISCHENABLAGE() eingestellt werden.

1.74 ABLAGESCHREIBEN([Block])

ABLAGESCHREIBEN([Block])

(Englisch: CLIPWRITE)

Schreibt den angegebene Block in die Zwischenablage. Wird der Parameter ausgelassen, so wird der aktuelle Block verwandt.

Die Einstellungen für den Zwischenablagezugriff können mit ZWISCHENABLAGE() eingestellt werden.

1.75 BLOCKSPEICHERN([Block];[Name])

BLOCKSPEICHERN([Block];[Name])

(Englisch: SAVEBLOCK)

Speichert den aktuellen bzw. mit Block angegebenen Block der aktuellen Tabelle als Name im Standard-TurboCalc-Format (siehe Anhang "Tabellenformate"). Fehlt Name, so wird der Datei-Requester zur Auswahl eines Namens angezeigt..

Block: Der Block, der gespeichert werden soll.

Name: Gibt den Namen der Datei an, unter dem die Tabelle gespeichert werden soll. Möglichst mit komplettem Pfad (etwa "DH0:TurboCalc/Tabellen/Test.TCD") angeben. Wird der Name ausgelassen, so erscheint der Datei-Requester.

Im Makromodus: Wird dieser Befehl im Makromodus benutzt, so wird der Zelle, in der dieser Befehl steht, der Wert WAHR zugewiesen, falls das Speichern erfolgreich war. Trat beim Speichern ein Fehler auf bzw. wurde das Speichern vom Benutzer per >Abbruch< am Dateirequester abgebrochen, so enthält diese Zelle den Wert FALSCH. Dies kann dann mit SPRINGEWENN abgefragt werden (siehe bei "Rückgabewerte von Befehlen" am Beginn der Befehlsübersicht).

Ohne Parameter entspricht dies dem Menüpunkt <Datei-Block speichern als-TurboCalc>.

Verwandte Befehle:

SPEICHERN , **SPEICHERNALS** , **CSVSPERICHERN** , **SYLKSPERICHERN**

1.76 CSVBLOCKSPEICHERN([Block];[Name];[Trennzeichen])

CSVBLOCKSPEICHERN([Block];[Name];[Trennzeichen])

(Englisch: CSVSAVEBLOCK)

Speichert den aktuellen bzw. mit Block angegebenen Block der aktuellen Tabelle als Name im CSV-Format (siehe Anhang "Tabellenformate"). Fehlt Name, so wird der Datei-Requester zur Auswahl eines Namens angezeigt..

Block: Der Block, der gespeichert werden soll.

Name: Gibt den Namen der Datei an, unter dem die Tabelle gespeichert werden soll. Möglichst mit komplettem Pfad (etwa "DH0:TurboCalc/Tabellen/Test.CSV") angeben. Wird der Name ausgelassen, so erscheint der Datei-Requester.

Trennzeichen: Gibt das Zeichen an, mit dem das CSV-Format die einzelnen Zellinhalte voneinander trennt. Fehlt diese Angabe, so wird das Semikolon ";" als Standard genommen. Das Zeichen muß in Anführungszeichen stehen, etwa ",", ";" oder " " (Leerzeichen). "T" ist dabei Tab, "S" das Leerzeichen (Space).

Im Makromodus: Wird dieser Befehl im Makromodus benutzt, so wird der Zelle, in der dieser Befehl steht, der Wert WAHR zugewiesen, falls das Speichern erfolgreich war. Trat beim Speichern ein Fehler auf bzw. wurde das Speichern vom Benutzer per >Abbruch< am Dateirequester abgebrochen, so enthält diese Zelle den Wert FALSCH. Dies kann dann mit SPRINGEWENN abgefragt werden (siehe bei "Rückgabewerte von Befehlen" am Beginn der Befehlsübersicht).

Ohne Parameter entspricht dies dem Menüpunkt <Datei-Block Speichern als CSV>.

Verwandte Befehle:

SPEICHERN , **SPEICHERNALS** , **SYLKSPEICHERN** , **CSVSPICHERN**

1.77 CSVEINFÜGEN([Block];[Name];[Trennzeichen])

CSVEINFÜGEN([Block];[Name];[Trennzeichen])

(Englisch: CSVINSERT)

Fügt die Tabelle Name an der Stelle Block ein. Die Tabelle muß dabei im CSV-Format sein - näheres zu diesem Format siehe "Tabellenformate" im Anhang.

Block: Bestimmt die linke obere Ecke, ab der die Daten eingefügt werden. Wird er nicht angegeben, so wird ab der aktuellen Cursorposition eingefügt.

Ohne Parameter entspricht dies dem Menüpunkt <Bearbeiten-Datei einfügen-CSV x>.

Im Makromodus: Wird dieser Befehl im Makromodus benutzt, so wird der Zelle, in der dieser Befehl steht, der Wert WAHR zugewiesen, falls das Einfügen erfolgreich war. Trat beim Einfügen ein Fehler auf bzw. wurde das Einfügen vom Benutzer per >Abbruch< am Dateirequester abgebrochen, so enthält diese Zelle den Wert FALSCH. Dies kann dann mit SPRINGEWENN abgefragt werden (siehe bei "Rückgabewerte von Befehlen" am Beginn der Befehlsübersicht). Einzelheiten siehe beim Befehl CSVLADEN (der bis auf den ersten Parameter gleich ist).

Verwandte Befehle:

TCDEINFÜGEN , **SYLKEINFÜGEN** , **PROCALCEINFÜGEN**

1.78 CSVLADEN([Name];[Trennzeichen])

CSVLADEN([Name];[Trennzeichen])

(Englisch: CSVLOAD)

Öffnet eine neue Tabelle (außer die aktuelle ist neu und noch leer) und lädt dann die Tabelle Name. Die Tabelle muß dabei im CSV-Format sein - näheres zu diesem Format siehe "Tabellenformate" im Anhang.

Name: Gibt den Namen der Datei an, die geladen werden soll. Möglichst mit komplettem Pfad (etwa "DH0:TurboCalc/Tabellen/Test.CSV") angeben. Wird der Name ausgelassen, so wird ein Datei-Auswahl-Fenster angezeigt, in dem der Benutzer die Datei auswählen kann.

Trennzeichen: Gibt das Zeichen an, mit dem das CSV-Format die einzelnen Zellinhalte voneinander trennt. Fehlt diese Angabe, so wird das Semikolon ";" als Standard genommen. Das Zeichen muß in Anführungszeichen stehen, etwa ",", ";" oder " " (Leerzeichen). "T" ist dabei Tab, "S" das Leerzeichen (Space).

Hinweis: Wird nur ein Parameter angegeben, so ist dies normalerweise der Name. Ausnahme: Hat der Name nur ein Zeichen, so wird es als Trennzeichen interpretiert und es erscheint der Datei-Requester, da ja kein Name vorhanden ist.

Im Makromodus: Wird dieser Befehl im Makromodus benutzt, so wird der Zelle, in der dieser Befehl steht, der Wert WAHR zugewiesen, falls das Laden erfolgreich war. Trat beim Laden ein Fehler auf bzw. wurde das Laden vom Benutzer per >Abbruch<

am Dateirequester abgebrochen, so enthält diese Zelle den Wert FALSCH. Dies kann dann mit SPRINGEWENN abgefragt werden (siehe bei "Rückgabewerte von Befehlen" am Beginn der Befehlsübersicht). Ohne Parameter entspricht dies dem Menüpunkt <Datei-Importieren-CSV>.

Verwandte Befehle:

LADEN , **SYLKLADEN** , **PROCALCLADEN**

1.79 CSVSPEICHERN([Name];[Trennzeichen])

CSVSPEICHERN([Name];[Trennzeichen])

(Englisch: CSVSAVE)

Speichert die aktuelle Tabelle als Name im CSV-Format (siehe Anhang "Tabellenformate"). Fehlt Name, so wird der Dateirequester zur Auswahl eines Namens angezeigt..

Name: Gibt den Namen der Datei an, unter dem die Tabelle gespeichert werden soll. Möglichst mit komplettem Pfad (etwa "DHO:TurboCalc/Tabellen/Test.CSV") angeben. Wird der Name ausgelassen, so erscheint der Dateirequester.

Trennzeichen: Gibt das Zeichen an, mit dem das CSV-Format die einzelnen Zelleninhalte voneinander trennt. Fehlt diese Angabe, so wird das Semikolon ";" als Standard genommen. Das Zeichen muß in Anführungszeichen stehen, etwa ",", ";" oder " " (Leerzeichen). "T" ist dabei Tab, "S" das Leerzeichen (Space).

Hinweis: Wird nur ein Parameter angegeben, so ist dies normalerweise der Name. Ausnahme: Hat der Name nur ein Zeichen, so wird es als Trennzeichen interpretiert und es erscheint der Dateirequester, da ja kein Name vorhanden ist.

Im Makromodus: Wird dieser Befehl im Makromodus benutzt, so wird der Zelle, in der dieser Befehl steht, der Wert WAHR zugewiesen, falls das Speichern erfolgreich war. Trat beim Speichern ein Fehler auf bzw. wurde das Speichern vom Benutzer per >Abbruch< am Dateirequester abgebrochen, so enthält diese Zelle den Wert FALSCH. Dies kann dann mit SPRINGEWENN abgefragt werden (siehe bei "Rückgabewerte von Befehlen" am Beginn der Befehlsübersicht).

Ohne Parameter entspricht dies dem Menüpunkt <Datei-Exportieren-CSV>.

Verwandte Befehle:

SPEICHERN , **SPEICHERNALS** , **SYLKSPEICHERN**

1.80 KONFIGLADEN()

KONFIGLADEN()

(Englisch: LOADCONFIG)

Entspricht dem Befehl <Optionen-Konfiguration-laden> und sucht die "Datei S:TurboCalc.CFG" und ändert die aktuellen Einstellungen entsprechend, falls die Datei gefunden wurde.

1.81 KONFIGSPEICHERN()

KONFIGSPEICHERN()

(Englisch: SAVECONFIG)

Entspricht dem Befehl <Optionen-Konfiguration-speichern> und speichert die aktuellen Einstellungen in die Datei "S:TurboCalc.CFG" - diese wird bei jedem Start von TurboCalc sowie jedem <Datei-Neu> automatisch gesucht und die jeweiligen Einstellungen übernommen.

Aktuelle Zelleninhalte werden durch diesen Befehl nicht mitgespeichert.

Tip: Bei dieser Datei handelt es sich um eine normale TurboCalc-Daten-Datei, wie sie durch Speichern erzeugt wird. Will man bei Datei-Neu auch Standard-Zelleninhalte bzw. Standard-Variablen definieren, so kann man dies in einer Datei tun und diese unter obigem Namen speichern (siehe auch SPEICHERN).

1.82 LADEN([Name])

LADEN([Name])

(Englisch: LOAD)

Öffnet eine neue Tabelle (außer die aktuelle ist neu und noch leer) und lädt dann die TurboCalc-Tabelle Name im Standardformat.

Name: Gibt den Namen der Datei an, die geladen werden soll. Möglichst mit komplettem Pfad (etwa "DH0:TurboCalc/Tabellen/Test.TC) angeben. Wird der Name ausgelassen, so wird ein Datei-Auswahl-Fenster angezeigt, in dem der Benutzer die Datei auswählen kann.

Im Makromodus: Wird dieser Befehl im Makromodus benutzt, so wird der Zelle, in der dieser Befehl steht, der Wert WAHR zugewiesen, falls das Laden erfolgreich war. Trat beim Laden ein Fehler auf bzw. wurde das Laden vom Benutzer per >Abbruch< am Dateirequester abgebrochen, so enthält diese Zelle den Wert FALSCH. Dies kann dann mit SPRINGEWENN abgefragt werden (siehe bei "Rückgabewerte von Befehlen" am Beginn der Befehlsübersicht).

Ohne Parameter entspricht dies dem Menüpunkt <Datei-Öffnen> .

Verwandte Befehle:

CSVLADEN , **SYLKLADEN** , **PROCALCLADEN**

1.83 PROCALCEINFÜGEN([Block];[Name])

PROCALCEINFÜGEN([Block];[Name])

(Englisch: PROCALCINSERT)

Fügt die Tabelle Name an der Stelle Block ein. Die Tabelle muß dabei im ProfessionalCalc-Dateiformat sein - näheres zu diesem Format siehe "Tabellenformate" im Anhang.

Block: Bestimmt die linke obere Ecke, ab der die Daten eingefügt werden. Wird er nicht angegeben, so wird ab der aktuellen Cursorposition eingefügt.

Im Makromodus: Wird dieser Befehl im Makromodus benutzt, so wird der Zelle, in der dieser Befehl steht, der Wert WAHR zugewiesen, falls das Einfügen erfolgreich war. Trat beim Einfügen ein Fehler auf bzw. wurde das Einfügen vom Benutzer per >Abbruch< am Dateirequester abgebrochen, so enthält diese Zelle den Wert FALSCH. Dies kann dann mit SPRINGEWENN abgefragt werden (siehe bei "Rückgabewerte von Befehlen" am Beginn der Befehlsübersicht).

Ohne Parameter entspricht dies dem Menüpunkt <Bearbeiten-Datei einfügen-ProCalc>.

Einzelheiten siehe beim Befehl PROCALCLADEN (der bis auf den ersten Parameter gleich ist).

Verwandte Befehle:

TCDEINFÜGEN , **CSVEINFÜGEN** , **SYLKEINFÜGEN**

1.84 PROCALCLADEN([Name])

PROCALCLADEN([Name])

(Englisch: PROCALCLOAD)

Öffnet eine neue Tabelle (außer die aktuelle ist neu und noch leer) und lädt dann die Tabelle Name. Die Tabelle muß dabei im Professional-Calc-Format sein - näheres zu diesem Format siehe "Tabellenformate" im Anhang.

Name: Gibt den Namen der Datei an, die geladen werden soll. Möglichst mit kompletten Pfad (etwa "DH0:TurboCalc/Tabellen/Test.PCF) angeben. Wird der Name ausgelassen, so wird ein Datei-Auswahl-Fenster angezeigt, in dem der Benutzer die Datei auswählen kann.

Im Makromodus: Wird dieser Befehl im Makromodus benutzt, so wird der Zelle, in der dieser Befehl steht, der Wert WAHR zugewiesen, falls das Laden erfolgreich war. Trat beim Laden ein Fehler auf bzw. wurde das Laden vom Benutzer per >Abbruch<

am Dateirequester abgebrochen, so enthält diese Zelle den Wert FALSCH. Dies kann dann mit SPRINGEWENN abgefragt werden (siehe bei "Rückgabewerte von Befehlen" am Beginn der Befehlsübersicht).

Ohne Parameter entspricht dies dem Menüpunkt <Datei-Importieren-Procalc>.

Verwandte Befehle:

LADEN , **CSVLADEN** , **SYLKLADEN**

1.85 SPEICHERN([Name])

SPEICHERN([Name])

(Englisch: SAVE)

Speichert die aktuelle Tabelle als Name im Standard TurboCalc-Format. Fehlt Name, so wird die Tabelle unter seinem aktuellen Namen gespeichert (falls es einen gibt, ansonsten erscheint der Datei-Requester zu Auswahl).

Name: Gibt den Namen der Datei an, unter dem die Tabelle gespeichert werden soll.. Möglichst mit komplettem Pfad (etwa "DH0:TurboCalc/Tabellen/Test.TCD") angeben. Wird der Name ausgelassen, so wird der aktuelle Name benutzt, falls vorhanden.

Im Makromodus: Wird dieser Befehl im Makromodus benutzt, so wird der Zelle, in der dieser Befehl steht, der Wert WAHR zugewiesen, falls das Speichern erfolgreich war. Trat beim Speichern ein Fehler auf bzw. wurde das Speichern vom Benutzer per >Abbruch< am Dateirequester abgebrochen, so enthält diese Zelle den Wert FALSCH. Dies kann dann mit SPRINGEWENN abgefragt werden (siehe bei "Rückgabewerte von Befehlen" am Beginn der Befehlsübersicht).

Ohne Parameter entspricht dies dem Menüpunkt <Datei-Speichern> .

Verwandte Befehle:

SPEICHERNALS , **CSVSPERICHERN** , **SYLKSPEICHERN**

1.86 SPEICHERNALS([Name])

SPEICHERNALS([Name])

(Englisch: SAVEAS)

Speichert die aktuelle Tabelle im Standard TurboCalc-Format. Es wird jedoch immer ein Datei-Requester angezeigt. Ist Name vorhanden, so wird dieser als Vorgabe benutzt, ansonsten dient der aktuelle Tabellename als Vorgabe.

Name: Gibt die Vorgabe für den Datei-Requester an. Möglichst mit komplettem Pfad (etwa "DH0:TurboCalc/Tabellen/Test.TCD") angeben.

Im Makromodus: Wird dieser Befehl im Makromodus benutzt, so wird der Zelle, in der dieser Befehl steht, der Wert WAHR zugewiesen, falls das Speichern erfolgreich war. Trat beim Speichern ein Fehler auf bzw. wurde das Speichern vom Benutzer per >Abbruch< am Dateirequester abgebrochen, so enthält diese Zelle den Wert FALSCH. Dies kann dann mit SPRINGEWENN abgefragt werden (siehe bei "Rückgabewerte von Befehlen" am Beginn der Befehlsübersicht).

Ohne Parameter entspricht dies dem Menüpunkt <Datei-Speichern als>.

Verwandte Befehle:

SPEICHERN , **CSVSPERICHERN** , **SYLKSPEICHERN**

1.87 SYLKBLOCKSPEICHERN([Block];[Name])

SYLKBLOCKSPEICHERN([Block];[Name])

(Englisch: SYLKSAVEBLOCK)

Speichert den aktuellen bzw. mit Block angegebenen Block der aktuellen Tabelle als Name im SYLK-Format (siehe Anhang "Tabellenformate"). Fehlt Name, so wird der Datei-Requester zur Auswahl eines Namens angezeigt..

Block: Der Block, der gespeichert werden soll.

Name: Gibt den Namen der Datei an, unter dem die Tabelle gespeichert werden soll. Möglichst mit komplettem Pfad (etwa "DH0:TurboCalc/Tabellen/Test.SLK") angeben. Wird der Name ausgelassen, so erscheint der Datei-Requester.

Im Makromodus: Wird dieser Befehl im Makromodus benutzt, so wird der Zelle, in der dieser Befehl steht, der Wert WAHR zugewiesen, falls das Speichern erfolgreich war. Trat beim Speichern ein Fehler auf bzw. wurde das Speichern vom Benutzer per >Abbruch< am Dateirequester abgebrochen, so enthält diese Zelle den Wert FALSCH. Dies kann dann mit SPRINGEWENN abgefragt werden (siehe bei "Rückgabewerte von Befehlen" am Beginn der Befehlsübersicht).

Ohne Parameter entspricht dies dem Menüpunkt <Datei-Block speichern als-SYLK>.

Verwandte Befehle:

SPEICHERN , **SPEICHERNALS** , **CSVSPERICHERN** , **SYLKSPEICHERN**

1.88 SYLKEINFÜGEN([Block];[Name])

SYLKEINFÜGEN([Block];[Name])

(Englisch: SYLKINSERT)

Fügt die Tabelle Name an der Stelle Block ein. Die Tabelle muß dabei im SYLK-Format sein - näheres zu diesem Format siehe "Tabellenformate" im Anhang.

Block: Bestimmt die linke obere Ecke, ab der die Daten eingefügt werden. Wird er nicht angegeben, so wird ab der aktuellen Cursorposition eingefügt.

Im Makromodus: Wird dieser Befehl im Makromodus benutzt, so wird der Zelle, in der dieser Befehl steht, der Wert WAHR zugewiesen, falls das Einfügen erfolgreich war. Trat beim Einfügen ein Fehler auf bzw. wurde das Einfügen vom Benutzer per >Abbruch< am Dateirequester abgebrochen, so enthält diese Zelle den Wert FALSCH. Dies kann dann mit SPRINGEWENN abgefragt werden (siehe bei "Rückgabewerte von Befehlen" am Beginn der Befehlsübersicht).

Ohne Parameter entspricht dies dem Menüpunkt <Bearbeiten-Datei einfügen-SYLK>.

Einzelheiten siehe beim Befehl SYLKLADEN (der bis auf den ersten Parameter gleich ist).

Verwandte Befehle:

TCDEINFÜGEN , **CSVEINFÜGEN** , **PROCALCEINFÜGEN**

1.89 SYLKLADEN([Name])

SYLKLADEN([Name])

(Englisch: SYLKLOAD)

Öffnet eine neue Tabelle (außer die aktuelle ist neu und noch leer) und lädt dann die Tabelle Name. Die Tabelle muß dabei im SYLK-Format sein - näheres zu diesem Format siehe "Tabellenformate" im Anhang.

Name: Gibt den Namen der Datei an, die geladen werden soll. Möglichst mit komplettem Pfad (etwa "DH0:TurboCalc/Tabellen/Test.SLK") angeben. Wird der Name ausgelassen, so wird ein Datei-Auswahl-Fenster angezeigt, in dem der Benutzer die Datei auswählen kann.

Im Makromodus: Wird dieser Befehl im Makromodus benutzt, so wird der Zelle, in der dieser Befehl steht, der Wert WAHR zugewiesen, falls das Laden erfolgreich war. Trat beim Laden ein Fehler auf bzw. wurde das Laden vom Benutzer per >Abbruch< am Dateirequester abgebrochen, so enthält diese Zelle den Wert FALSCH. Dies kann dann mit SPRINGEWENN abgefragt werden (siehe bei "Rückgabewerte von Befehlen" am Beginn der Befehlsübersicht).

Ohne Parameter entspricht dies dem Menüpunkt <Datei-Importieren-Sylk>.

Verwandte Befehle:

LADEN , **CSVLADEN** , **PROCALCLADEN**

1.90 SYLKSPEICHERN([Name])

SYLKSPEICHERN([Name])

(Englisch: SYLKSAVE)

Speichert die aktuelle Tabelle als Name im SYLK-Format (siehe Anhang "Tabellenformate"). Fehlt Name, so wird der Dateirequester zur Auswahl eines Namens angezeigt..

Name: Gibt den Namen der Datei an, unter dem die Tabelle gespeichert werden soll.. Möglichst mit komplettem Pfad (etwa "DHO:TurboCalc/Tabellen/Test.SLK") angeben. Wird der Name ausgelassen, so erscheint der Dateirequester.

Im Makromodus: Wird dieser Befehl im Makromodus benutzt, so wird der Zelle, in der dieser Befehl steht, der Wert WAHR zugewiesen, falls das Speichern erfolgreich war. Trat beim Speichern ein Fehler auf bzw. wurde das Speichern vom Benutzer per >Abbruch< am Dateirequester abgebrochen, so enthält diese Zelle den Wert FALSCH. Dies kann dann mit SPRINGEWENN abgefragt werden (siehe bei "Rückgabewerte von Befehlen" am Beginn der Befehlsübersicht).

Ohne Parameter entspricht dies dem Menüpunkt <Datei-Exportieren-SYLK>.

Verwandte Befehle:

SPEICHERN , **SPEICHERNALS** , **CSV SPEICHERN**

1.91 TCDEINFÜGEN([Block];[Name])

TCDEINFÜGEN([Block];[Name])

(Englisch: TCDINSERT)

Fügt die TurboCalc-Tabelle Name im Standardformat an der Stelle Block ein. (TCD steht hier für TCDData)

Block: Bestimmt die linke obere Ecke, ab der die Daten eingefügt werden. Wird er nicht angegeben, so wird ab der aktuellen Cursorposition eingefügt.

Im Makromodus: Wird dieser Befehl im Makromodus benutzt, so wird der Zelle, in der dieser Befehl steht, der Wert WAHR zugewiesen, falls das Einfügen erfolgreich war. Trat beim Einfügen ein Fehler auf bzw. wurde das Einfügen vom Benutzer per >Abbruch< am Dateirequester abgebrochen, so enthält diese Zelle den Wert FALSCH. Dies kann dann mit SPRINGEWENN abgefragt werden (siehe bei "Rückgabewerte von Befehlen" am Beginn der Befehlsübersicht).

Ohne Parameter entspricht dies dem Menüpunkt <Bearbeiten-Datei einfügen-TurboCalc>.

Einzelheiten siehe beim Befehl LADEN (der bis auf den ersten Parameter gleich ist).

Verwandte Befehle:

CSVEINFÜGEN , **SYLKEINFÜGEN** , **PROCALCEINFÜGEN**

1.92 Datenbank

Datenbank

DATENBANK([Bereich])

DBEXTRAHIEREN([Zelle])

DBLETZTER(Zelle)

DBLÖSCHEN()

DBSORTIEREN(Aufsteigend;[Zelle])

DBSUCHEN([Zelle])

KRITERIEN([Bereich])

1.93 DATENBANK([Bereich])

DATENBANK([Bereich])

(Englisch: DATABASE)

Bestimmt den neuen Datenbankbereich (entspricht <Daten-Datenbank definieren>).

Wird Bereich nicht angegeben, so wird der aktuell markierte Block benutzt.

Näheres zum Datenbankbereich siehe Abschnitt "Datenbank"

Beispiel:

DATENBANK() bestimmt den aktuellen Block als neuen Datenbankbereich

DATENBANK(A1:C10) der Block A1:C10 ist neuer Datenbankbereich

Verwandte Befehle:

KRITERIEN , **DBSUCHEN** , **DBEXTRAHIEREN** , **DBLÖSCHEN** , **DBSORTIEREN**

1.94 DBEXTRAHIEREN([Zelle])

DBEXTRAHIEREN([Zelle])

(Englisch: DBEXTRACT)

Kopiert alle Datensätze die den aktuellen Suchkriterien entsprechen (entspricht <Daten-Suchen und Kopieren>).

Zelle bestimmt die Position, an die die Daten kopiert werden sollen. Achten Sie auf genügend Platz, so daß nichts zufällig überschrieben wird!

Fehlt Zelle, so wird ab der aktuellen Cursorposition eingefügt.

Näheres dazu siehe Abschnitt "Datenbank"

Beispiel:

DBEXTRAHIEREN(A100) kopiert alle passenden Sätze nach A100.

Verwandte Befehle:

DATENBANK , **KRITERIEN** , **DBSUCHEN** , **DBLÖSCHEN** , **DBSORTIEREN**

1.95 DBLETZTER(Zelle)

DBLETZTER(Zelle)

(Engl: DBPREV)

Dieser Befehl arbeitet wie DBSUCHEN, jedoch ist die Suchreihenfolge von unten nach oben!

Zelle: Dies bestimmt die Position, ab der die Suche begonnen werden soll. Fehlt die Angabe, so wird die aktuelle Cursorposition benutzt. Ist diese Angabe außerhalb des Datenbankbereichs, so wird in der letzten Datenbankzeile begonnen.

DBLETZTER gibt folgenden Rückgabewert zurück (d.h. die Zelle, in der der Makrobefehl steht, hat folgenden Inhalt):

TRUE: es wurde eine Zeile gefunden, diese ist markiert

FALSE: keine Übereinstimmung.

Anmerkung: Im Gegensatz zu DBSUCHEN gibt DBLETZTER nie eine Meldung aus (etwa "kein Datensatz gefunden"), auch nicht im Direktmodus!

1.96 DBLÖSCHEN()

DBLÖSCHEN()

(Englisch: DBDELETE)

Löscht alle Datensätze die den aktuellen Suchkriterien entsprechen (entspricht <Daten-Löschen>).

Näheres dazu siehe Abschnitt "Datenbank"

Beispiel:

DBLÖSCHEN() löscht alle Zeilen, die den Suchkriterien entsprechen.

Verwandte Befehle:

DATENBANK , **KRITERIEN** , **DBSUCHEN** , **DBEXTRAHIEREN** , **DBSORTIEREN**

1.97 DBSORTIEREN(Aufsteigend;[Zelle])

DBSORTIEREN(Aufsteigend;[Zelle])

(Englisch: DBSORT)

Sortiert alle Datensätze aufsteigend nach der aktuellen Spalte (entspricht <Daten-Bereich sortieren>).

Aufsteigend: Sortiert die Daten aufsteigend, falls dieser Parameter ungleich 0 ist oder fehlt. Bei 0 bzw. FALSCH wird absteigend sortiert.

Zelle bestimmt die Spalte, nach der sortiert werden soll. Fehlt Zelle, so wird nach der aktuellen Cursorposition sortiert. (Liegt die Spalte, nach der sortiert werden soll, außerhalb des Datenbankbereichs, so wird nichts sortiert!)

Näheres dazu siehe Abschnitt "Datenbank"

Beispiel:

DBSORT(A1) sortiert nach der ersten Spalte (sofern der Datenbankbereich bei Spalte A beginnt).

Verwandte Befehle:

DATENBANK , **KRITERIEN** , **DBSUCHEN** , **DBEXTRAHIEREN** , **DBLÖSCHEN**

1.98 DBSUCHEN([Zelle])

DBSUCHEN([Zelle])

(Englisch: DBFIND)

Sucht nach der nächsten Übereinstimmung der Datenbank (entspricht <Daten-Suchen>).

Zelle bestimmt die Position, ab der die Suche begonnen werden soll. (A1 bedeutet immer ab Anfang)

Fehlt Zelle, so wird ab der aktuellen Cursorposition gesucht.

Im Makromodus: Wird dieser Befehl im Makromodus benutzt, so wird der Zelle, in der dieser Befehl steht, der Wert WAHR zugewiesen, falls die Suche erfolgreich war. Wurde nichts gefunden, so enthält diese Zelle den Wert FALSCH. Dies kann dann mit SPRINGEWENN abgefragt werden (siehe Beispiel unten). (Im Makromodus erscheint dann natürlich kein Fenster, um den Benutzer darauf aufmerksam zu machen, daß nichts gefunden wurde - dies muß dann ggf. innerhalb des Makros geschehen.)

Näheres dazu siehe Abschnitt "Datenbank"

Beispiel:

DBSUCHEN(A1) sucht die erste Zeile, die den Suchkriterien entspricht.

DBSUCHEN() setzt diese Suche fort.

A10 =DBSUCHEN(A1)
 A11 =SPRINGEWENN(#A10;A14)
 A12 =MELDUNG("Nichts gefunden")
 A13 =SPRINGE(A30).
 A14 hier beginnt das eigentliche Programm
 ...
 A30 =RETURN()

Verwandte Befehle:

DATENBANK , KRITERIEN , DBEXTRAHIEREN , DBLÖSCHEN , DBSORTIEREN

1.99 KRITERIEN([Bereich])

KRITERIEN([Bereich])

(Englisch: CRITERIA)

Bestimmt den neuen Suchkriterienbereich (entspricht <Daten-Suchkriterien definieren>).

Wird Bereich nicht angegeben, so wird der aktuell markierte Block benutzt.

Näheres zum Suchkriterienbereich siehe Abschnitt "Datenbank"

Beispiel:

KRITERIEN() bestimmt den aktuellen Block als neue Suchkriterien

KRITERIEN(A1:C10) der Block A1:C10 ist neuer Suchkriterienbereich

Verwandte Befehle:

DATENBANK , DBSUCHEN , DBEXTRAHIEREN , DBLÖSCHEN , DBSORTIEREN

1.100 Optionen

Optionen

BILDAUFBAU(Modus)

DARSTELLUNG(Titel;Raster;Werkzeugl.;Formeln;Null;Cursorm,..)

DEFPUBSCREEN(Bildschirm)

DRUCKBEREICH(Aktivieren;[Bereich])

DRUCKFORMAT(LR;RR;OR;UR;Stil;KZeile;KText;FZeile;FText;Titel;Raster)

GLOBALEOPTIONEN()

LADEOPTIONEN(KeinAutoMakro;Verbergen;Passwort)

LÄNDEROPTIONEN(ZF1;ZF2;DF;Währung;WPräfix;WSuffix;Inch)

PAPIERFORMAT()

SCHUTZOPTIONEN()

SEITENVORSCHUB(Flag)

SHANGHAI(Modus)

SMARTREFRESH(Flag)

TABELLENOPTIONEN(MaxBreite;MaxHöhe;Berechnung;Return;Richtung;Icons)

TASTE(Taste;Befehl)

ZWISCHENABLAGE(Einheit;Trennzeichen;Anführungszeichen)

ZWISCHENSPEICHERN(Flag)

1.101 BILDAUFBAU(Modus)

BILDAUFBAU(Modus)

(Englisch: REFRESH)

Hiermit kann der Bildaufbau aus- bzw. eingeschaltet sowie ein neuer Bildaufbau erzwungen werden.

Modus:

0 (bzw. FALSCH) schaltet den Bildaufbau aus.

1 (bzw. WAHR) schaltet den Bildaufbau ein.

2 erzwingt einen neuen Bildaufbau. Dies ist zu empfehlen, wenn der Bildaufbau ausgeschaltet war und nun wieder eingeschaltet wurde.

Hinweis: Diese Funktion schaltet sowohl den Bildaufbau als auch die automatische Berechnung aus. Dadurch werden Makros schneller abgearbeitet. Sollten dann die Formeln neu berechnet werden, so ist ein NEUBERECHNEN auszuführen.

Hinweis: Dieser Befehl wirkt nur auf Makro- und ARexxbefehle. Bei normalen Eingaben bzw. Befehlsaufrufen wird weiterhin das Bild korrekt erneuert. Trotzdem sollte am Ende eines Makros der Bildaufbau mit BILDAUFBAU(WAHR) wieder geschaltet werden, um ggf. dann mit BILDAUFBAU(2) den Bildschirm zu erneuern.

Beispiel:

Folgender Makroausschnitt beschleunigt den Aufbau:

=BILDAUFBAU(0)

... hier kommen die Befehle, die schnell ausgeführt werden sollen

=BILDAUFBAU(1)

=NEUBERECHNEN()

=BILDAUFBAU(2)

=RÜCKSPRUNG()

1.102 DARSTELLUNG(Titel;Raster;Werkzeugl.;Formeln;Null;Cursorm,,)

DARSTELLUNG(Titel;Raster;Werkzeugl.;Formeln;Null;Cursorm,,)

(Englisch: DISPLAY)

Dieser Befehl entspricht (ohne Parameter) dem Menüpunkt <Optionen-Darstellung...> und erlaubt so die Einstellung der Anzeige-Parameter eines TurboCalc-Fensters.

Hinweis: Die Darstellungsoptionen beziehen sich nur auf das aktuelle Fenster - sie können (und müssen) für jedes Fenster separat eingestellt werden!

Titel: Soll die Zeilen- und Spaltentitelzeile angezeigt werden?

Raster: Soll das Raster eingeblendet werden?

Werkzeuggestreife: Soll die Symbolleiste angezeigt werden?

Formeln: Sollen Formeln (statt den Werten) angezeigt werden?

Null: Sollen Nullwerte angezeigt werden?

Cursormodus: Bestimmt, ob der Cursor als Rahmen oder als Block gezeichnet werden soll.

Für jeden Parameter kann folgendes angegeben werden:

0 bzw. FALSCH: Nein

1 bzw. WAHR: Ja

2: wechseln, d.h. war der Parameter vorher gesetzt, so wird er gelöscht und umgekehrt
-1 bzw. auslassen des Parameters: die Einstellung bleibt unverändert.

Bei jedem anderen Wert erscheint die Fehlermeldung "Falscher Parameter..."

Beispiel:

DARSTELLUNG(;;2)

Dieser Befehl wechselt zwischen Darstellung von Werten und von Formeln.

1.103 DEFPUBSCREEN(Bildschirm)

DEFPUBSCREEN(Bildschirm)

(Erst ab OS2.0!)

Bestimmt den aktuellen DefaultPublicScreen, d.h. den Bildschirm auf dem die Meldungen sowie weitere Fenster erscheinen sollen. Näheres dazu entnehmen Sie bitte Ihrem Benutzerhandbuch.

Bildschirm: Ist der Name des PublicScreens (etwa "WorkBench" für die WorkBench). Wird kein Name angegeben, so wird der TurboCalc-Bildschirm benutzt (falls dieser geöffnet ist).

Beispiel:

DEFPUBSCREEN() wählt den TurboCalc-Bildschirm als neuen DefaultPublicScreen

DEFPUBSCREEN("WorkBench") der WorkBench-Bildschirm ist wieder der neuen Standard-Public-Screen

Verwandte Befehle:

SHANGHAI

1.104 DRUCKBEREICH(Aktivieren;[Bereich])

DRUCKBEREICH(Aktivieren;[Bereich])

(Englisch: PRINTRANGE)

Dies bestimmt den Bereich, der beim Drucken gedruckt werden soll. (Entsprechend den Menüpunkten <Optionen-Druckbereich> .

Aktivieren: Bestimmt, ob der Druckbereich eingeschaltet werden soll. Bei 0 bzw. FALSCH wird er ausgeschaltet und die ganze Tabelle zum Ausdrucken vorgemerkt..

Bereich: Bestimmt den Bereich, der als Druckbereich markiert werden soll. (Nur sinnvoll, falls Aktivieren auf einen Wert ungleich Null gesetzt wurde). Wird dieser Bereich ausgelassen, so wird der aktuelle Block markiert.

Beispiel:

DRUCKBEREICH()

markiert den aktuellen Block als neuen Druckbereich.

DRUCKBEREICH(0)

schaltet einen ggf. vorhandenen Druckbereich aus.

DRUCKBEREICH(;A1:C5)

DRUCKBEREICH(WAHR;A1:C5)

Diese beiden Befehle legen den Block A1 bis C5 als neuen Druckbereich fest.

1.105 DRUCKFORMAT(LR;RR;OR;UR;Stil;KZeile;KText;FZeile;FText;Titel;Raster)

DRUCKFORMAT(LR;RR;OR;UR;Stil;KZeile;KText;FZeile;FText;Titel;Raster)

(Englisch: PRINTFORMAT)

Hiermit kann das Druckformat bestimmt werden.

LR: Dies ist der linke Rand in cm

RR: Der rechte Rand in cm

OR: Der obere Rand in cm

UR: Der untere Rand in cm

Stil: 0=Pica, 1=Elite, 2=Condensed

KZeile: Bestimmt, ob eine Kopfzeile gedruckt werden soll. WAHR bzw. Werte ungleich Null bedeuten ja, bei FALSCH wird sie nicht gedruckt.

KText: Bestimmt den Text der als Kopfzeile gedruckt werden soll, falls Kopfzeile auf WAHR steht. Der Text muß in Anführungszeichen stehen, bzw. es muß sich um eine Formel handeln, die einen Text zurückgibt (etwa LINKS...)

FZeile: Bestimmt, ob eine Fußzeile gedruckt werden soll. WAHR bzw. Werte ungleich Null bedeuten ja, bei FALSCH wird sie nicht gedruckt.

FText: Bestimmt den Text der als Fußzeile gedruckt werden soll, falls Kopfzeile auf WAHR steht. Der Text muß in Anführungszeichen stehen, bzw. es muß sich um eine Formel handeln, die einen Text zurückgibt (etwa LINKS...)

Titel: Bestimmt, ob Zeilen- und Spaltentitel gedruckt werden sollen. WAHR bzw. Werte ungleich Null bedeuten ja, bei FALSCH wird sie nicht gedruckt.

Raster: Bestimmt, ob der Ausdruck mit einem Raster erfolgen soll. WAHR bzw. Werte ungleich Null bedeuten ja, bei FALSCH wird nicht gedruckt.

Beispiel:

DRUCKFORMAT(2;2;2;2;0;WAHR;"Kopfzeile";FALSCH;"Fußzeile";FALSCH;FALSCH)

Bestimmt das Druckformat wie folgt: Ränder je 2 cm, Pica, Kopfzeile ein (Text: "Kopfzeile"), Fußzeile aus (Text: "Fußzeile"). Weder Titel noch Raster.

DRUCKFORMAT(;;;;;;WAHR)

Schaltet die Fußzeile auch ein, ändert sonst nichts

DRUCKFORMAT(;;;;;;"Seite %S")

Ändert die Fußzeile.

1.106 GLOBALEOPTIONEN()

GLOBALEOPTIONEN()

(Englisch: GLOBALFLAGS)

Entspricht <Optionen-Globale Einstellungen>.

1.107 LADEOPTIONEN(KeinAutoMakro;Verbergen;Passwort)

LADEOPTIONEN(KeinAutoMakro;Verbergen;Passwort)

(Englisch: OPENFLAGS)

KeinAutoMakro: Steht dieses Flag auf WAHR, so wird ein ggf. vorhandenes AUTOMATIK (bzw. AUTOMATIC)-Makro nicht ausgeführt. Dies ist praktisch, wenn man eine Tabelle zum Bearbeiten öffnen möchte, ohne daß dabei dieses Makro ausgeführt wird.

Verbergen: Ist dieses Flag angewählt (WAHR), so wird das Fenster der zu ladenden Tabelle nicht geöffnet, sondern gleich als 'verborgener' markiert. (Es kann dann später mittels <Optionen-Tabelle einblenden> angezeigt werden.) Diese Funktion kann von Makros benutzt werden, um Tabellen (für den Benutzer fast unmerklich) im Hintergrund nachzuladen.

Passwort: Dies speichert ein 'Vorgabe-Passwort', daß benutzt wird, falls die Tabelle passwortgeschützt ist. Diese Passwort-Vorgabe bleibt bis zum nächsten ändern durch diesen Befehl erhalten. Mit "" (leerer Text) schalten Sie die Passwort-Vorgabe wieder aus. (Stimmt das so angegebene Passwort nicht mit dem Tabellen-Passwort überein, so erscheint weiterhin ein Requester, in dem der Benutzer nach dem richtigen Passwort gefragt wird.)

Dies kann praktisch sein, falls Sie aus einem Makro heraus passwortgeschützte Tabellen öffnen möchten (etwa mehrere). Dann muß der Benutzer nur einmal nach seinem Passwort gefragt werden und dieses kann dann auf alle Tabellen angewendet werden.

Hinweis: Durch Anwendung dieser Option geht natürlich die Wirkung des Passwortschutzes eingeschränkt und die Sicherheit reduziert. Man sollte sich beim Anwenden dieses Befehls darüber bewußt sein. (Beispiel: Der Benutzer gibt im Makro sein Passwort ein, dieses wird als Vorgabepasswort gesetzt. Der Benutzer verläßt kurz den Raum. Nun kann jeder das Makro unterbrechen und beliebige, mit dem gleichen Passwort geschützte Tabellen laden!)

1.108 LÄNDEROPTIONEN(ZF1;ZF2;DF;Währung;WPräfix;WSuffix;Inch)

LÄNDEROPTIONEN(ZF1;ZF2;DF;Währung;WPräfix;WSuffix;Inch)

(Englisch: LOCALE)

Ohne Parameter entspricht dies dem Menüpunkt <Optionen-Nationales...> und erlaubt die Festlegung länderspezifischer Optionen.

Mit Parameter können diese direkt geändert werden:

ZF1: Bestimmt das Trennzeichen zwischen Vorkomma- und Nachkommastellen:

0: Punkt (1.23)

1: Komma (1,23)

ZF2: Bestimmt das Tausendertrennzeichen:

0: Komma (1,200)

1: Punkt (1.200)

2: Apostroph (1'200)

3: Strich (1_200)

4: Leerzeichen (1 200)

DF bestimmt das Datumsformat:

0: Punkt (30.09.93)

1: Strich (30-09-93)

2: Schrägstrich (30/09/93)

Währung bestimmt das Währungssymbol:

0: DM

1: Dollar

2: Benutzerbestimmung (siehe nächsten beiden)

3: Pfund

4: Yen

5: FF (Franc)

6: ÖS (Schilling)

7: SFr (Franken)

WPräfix ist der Text, der vor der Zahl stehen soll

WSuffix ist der Text, der nach der Zahl stehen soll.

(Einer oder beide können auch leer sein, dann "" als Parameter angeben).

Inch bestimmt, ob Längenangaben in "cm" (0 oder FALSCH) oder aber in "inch" (1 oder WAHR) eingegeben und angezeigt werden sollen. Dies bezieht sich jedoch nur auf die Requester, für die Makrobefehle müssen die Längenangaben immer in cm eingegeben werden (um Kompatibilität zu gewährleisten).

1.109 PAPIERFORMAT()

PAPIERFORMAT()

(Englisch: PAPERFORMAT)

Dies entspricht <Format-Papierformat>.

1.110 SCHUTZOPTIONEN()

SCHUTZOPTIONEN()

(Englisch: PROTECTFLAGS)

Entspricht <Optionen-Schutzmerkmale...> .

1.111 SEITENVORSCHUB(Flag)

SEITENVORSCHUB(Flag)

(Englisch: FORMFEED)

Hiermit kann beim bestimmt werden, ob beim Ausdruck am Seitenende das Steuerzeichen Formfeed (12) statt mehreren Zeilenvorschüben benutzt werden soll.

Normalerweise druckt TurboCalc entsprechend der Seitenlänge mehrere Zeilenvorschübe, bis die Seite zu enden ist. Dadurch kann das Papierformat beliebig bestimmt werden.

Ist "FormFeed" eingeschaltet, so wird statt diesen Zeilenvorschüben ein Seitenvorschub gedruckt. Dadurch muß am Drucker die Seitenhöhe eingestellt sein.

Flag: 0/FALSCH = kein "Formfeed"

1/WAHR = "Formfeed"

Tip: Bei Ganzseitendruckern (Laser- bzw. Tintenstrahldruckern) kann das Einschalten dieser Option sinnvoll sein.

Hinweis: Auch wenn diese Option eingeschaltet ist, muß das Papierformat in TurboCalc bestimmt werden, da TurboCalc damit die Anzahl der druckbaren Zeilen bestimmt.

1.112 SHANGHAI(Modus)

SHANGHAI(Modus)

(Erst ab OS2.0!)

Dies schaltet den Shanghai-Modus ein bzw. aus:

Modus: Ist die Summe aus folgenden beiden Einstellungen:

1: SHANGHAI (d.h. alle neu geöffneten WorkBench-Fenster erscheinen auf dem TurboCalc Bildschirm)

2: POPPUBSCREEN (d.h. für Requester, die auf dem TurboCalc-Bildschirm erscheinen, wird dieser auch nach vorne geholt)

Mit SHANGHAI(3) wird beides eingeschaltet.

1.113 SMARTREFRESH(Flag)

SMARTREFRESH(Flag)

Hiermit wird das SmartRefresh an bzw. ausgeschaltet. Ist es aktiviert, so werden die Fensterinhalte intern zwischengespeichert, sobald ein Fenster von einem anderen überdeckt wird. Dadurch ist nicht mehr so oft ein neuer Bildaufbau nötig und die Geschwindigkeit wird entsprechend erhöht. Nachteil ist jedoch der erhöhte Speicherverbrauch für das Zwischenspeichern der Fensterinhalte.

Flag:

0 Smartrefresh wird ausgeschaltet

1 Smartrefresh nur für Tabellenfenster

2 Smartrefresh nur für Diagrammfenster

3 Smartrefresh für alle Fenster

Ohne Parameter entspricht dies dem Befehl <Optionen-Bildschirm-Bildaufbau>.

1.114 TABELLENOPTIONEN(MaxBreite;MaxHöhe;Berechnung;Return;Richtung;Icons)

TABELLENOPTIONEN(MaxBreite;MaxHöhe;Berechnung;Return;Richtung;Icons)

(Englisch: SHEETFLAGS)

Hiermit werden die Tabellen-Parameter bestimmt. (Ohne Parameter entspricht dies dem Menüpunkt <Optionen-Tabelle> .

MaxBreite: Bestimmt die maximale Breite der Tabelle (zwischen 50 und 700!).

MaxHöhe: Bestimmt die maximale Höhe der Tabelle (zwischen 50 und 9999999).

Berechnung: Bestimmt die Art der Berechnung (Automatisch/Manuell). WAHR bzw. Werte ungleich Null bedeuten automatische Neuberechnung, bei FALSCH wird sie nicht automatisch durchgeführt.

Return: Bestimmt, ob bei <Return> der Cursor bewegt werden soll. WAHR bzw. Werte ungleich Null bedeuten ja, bei FALSCH wird er nicht bewegt.

Richtung: Gibt die Richtung an, in die der Cursor bewegt werden soll: 0=unten, 1=oben, 2=links, 3=rechts.

Icons: Bestimmt, ob beim Speichern der Tabelle ein Icon erzeugt werden soll. WAHR bzw. Werte ungleich Null bedeuten ja, bei FALSCH wird sie nicht gedruckt.

1.115 TASTE(Taste;Befehl)

TASTE(Taste;Befehl)

(Englisch: KEY)

Dieses Makro ermöglicht das Belegen bzw. Umdefinierten von beliebigen Tasten mit TurboCalc-Befehlen. Mögliche Einsatzgebiete sind dabei TurboCalc.STD (zum Festlegen beliebiger ShotCuts, die immer vorhanden sein sollen) oder aber in Makrobefehlen um etwa spezielle Makroprogramme durch einen Tastedruck aufrufen zu können.

Taste: Ist ein Text, der eine Taste angibt (siehe Tabelle unten).

Befehl: Ist der Befehl (Makro- oder Arexxbefehl), der ausgeführt wird, sobald die Taste gedrückt wird. Befehl kann auch ausgelassen werden, dann wird diese Taste wieder freigegeben. (Für "Standard-F2" den Befehl EDIT und für "Standard-F10" COMMAND benutzen)

Taste kann mit folgenden "Umschalttasten" beginnen (und jeweils mit "-" enden):

LSHIFT- RSHIFT- SHIFT- S-

CAPS-

CTRL- C-

LALT- RALT- ALT- A-

LAMIGA- RAMIGA- AMIGA-

Dabei steht L... bzw R... für genau die rechte oder die linke Taste. Die Bezeichnungen ohne L bzw. R bedeuten, daß es reicht, wenn eine Taste gedrückt ist. (Wird sowohl Lxxx als auch Rxxx angegeben, so entspricht dies xxx, es reicht dann also, eine Umschalt-Taste zu drücken.

Es können natürlich mehrere Umschalttasten kombiniert werden.

Anschließend (oder falls keine Umschalttasten benutzt werden auch ab Beginn) folgt dann eine der folgenden Tasten:

F1 ... F10

ESC

TAB

RETURN

DEL

BS

HELP

UP HOCH

DOWN RUNTER

LEFT LINKS

RIGHT RECHTS

Und für den Nummernblock:

NUM0..NUM9

ENTER

NUMA..NUMG (die restlichen Tasten des Nummernblocks, auf der deutschen Tastatur "[]/*-+.")

Oder es kann eine ganz normale Taste angefügt werden (etwa ctrl-a, alt-r)

Achtung: Nur normale Tasten (d.h. die ohne Umschalttasten erreicht werden können) verwenden. Für "!" etwa "shift-1" schreiben (auf der deutschen Tastatur)

LSHIFT-F1 ist die Taste F1, sofern die linke Shift-Taste gedrückt wird.

SHIFT-RALT-F1 eine beliebige Shift-Taste sowie die rechte Alt-Taste muß gedrückt sein, wenn F1 gedrückt wird.

Beispiele:

KEY("F1";"EDIT") die Taste F1 hat nun die Funktion, die normalerweise F2 hat. (Editierung einschalten)

KEY("F1") wieder die Standardbelegung für F1 herstellen

KEY("CTRL-X";"CUT") den Tasten Ctrl-X/C/V die Blockbefehle zuweisen

KEY("CTRL-C";"COPY") (normalerweise per r.Amiga erreichbar)

KEY("CTRL-V";"PASTE")

KEY("CTRL-A";"MACROPLAY(A1)") wird Ctrl-A gedrückt, so wird das Makro ab der Position A1 ausgeführt.

1.116 ZWISCHENABLAGE(Einheit;Trennzeichen;Anführungszeichen)

ZWISCHENABLAGE(Einheit;Trennzeichen;Anführungszeichen)

(Englisch: CLIPBOARD)

Hiermit können die Parameter für die Zwischenablage eingestellt werden. Wird kein Parameter angegeben, so wird entspricht dies <Bearbeiten-Clipboard-Einstellungen>. Eine ausführlichere Beschreibung der Parameter finden Sie dort.

Einheit: Zahl von 0 bis 255. (Std: 0)

Trennzeichen: Art des Trennzeichens, Zahl von 0 bis 4 (Std: 0)

0 Tab

1 Komma

2 Semikolon

3 Leerzeichen

4 mehrere Leerzeichen

Anführungszeichen: Zahl von 0 bis 2: (Std: 0)

0 nie Anführungszeichen

1 Anführungszeichen nur falls nötig

2 immer Anführungszeichen

Wird ein Parameter ausgelassen bzw. -1 angegeben, so wird die jeweilige Einstellung nicht geändert

Die Zwischenablage wird dann mit den Befehlen ABLAGELESEN und ABLAGESCHREIBEN angesprochen!

Beispiele

ZWISCHENABLAGE(1)

wechselt zur Zwischenablage-Nr. 1 - die anderen Parameter werden nicht geändert

CLIPBOARD(;0;0)

Tab, nie Anführungszeichen - die Zwischenablageeinheit wird nicht geändert.

1.117 ZWISCHENSPEICHERN(Flag)

ZWISCHENSPEICHERN(Flag)

(Englisch: AUTOSAVE)

Dies entspricht <Optionen-Autom. Speichern> und ruft das Fenster zur Parametereinstellung auf.

Wird Flag angegeben, so wird stattdessen die aktuelle Tabelle zwischengespeichert.

1.118 Menübefehle

Menübefehle

LÖSCHEMENÜPUNKT(Titel;Punkt)

LÖSCHEMENÜTITEL(Titel)

LÖSCHEMENÜUNTER(Titel;Punkt;Unter)

NEUERMENÜPUNKT(Name;Befehl;[Titel;Punkt])

NEUERMENÜTITEL(Name;[Titel])

NEUERMENÜUNTER(Name;Befehl;[Titel;Punkt;Unter])

NEUESMENÜ()

ZEIGEMENÜ()

1.119 LÖSCHEMENÜPUNKT(Titel;Punkt)

LÖSCHEMENÜPUNKT(Titel;Punkt)

(Englisch: DELMENUITEM)

Löscht den Menüpunkt an der Position Titel, Punkt.

Titel: Menütitel-Nummer, aus dem ein Menüpunkt gelöscht werden soll.

Punkt: Der zu löschende Punkt (0=erster...)

Beispiel:

LÖSCHEMENÜPUNKT(0;1) löscht, bei den Standard-Menüs, den Menüpunkt <Datei-Öffnen...>

1.120 LÖSCHEMENÜTITEL(Titel)

LÖSCHEMENÜTITEL(Titel)

(Englisch: DELMENUTITLE)

Löscht den Menütitel an der Position Titel.

Titel: . Menütitel-Nummer der gelöscht werden soll.(0=Anfang, 1= nach erster Position...)

Beispiel:

LÖSCHEMENÜTITEL(0) löscht, bei den Standard-Menüs, den ganzen Menütitel "Datei"

1.121 LÖSCHEMENÜUNTER(Titel;Punkt;Unter)

LÖSCHEMENÜUNTER(Titel;Punkt;Unter)

(Englisch: DELMENUSUB)

Löscht den Menüpunkt an der Position Titel, Punkt, Unter.

Titel: Menütitel-Nummer, aus dem ein Menüunterpunkt gelöscht werden soll.

Punkt: Menüpunkt-Nummer, aus dem ein Menüunterpunkt gelöscht werden soll.

Unter: Der zu löschende Unterpunkt (0=erster...)

Beispiel:

LÖSCHEMENÜPUNKT(0;5;0) löscht, bei den Standard-Menüs, den Menüpunkt <Datei-Importieren-Procalc>

1.122 NEUERMENÜPUNKT(Name;Befehl;[Titel;Punkt])

NEUERMENÜPUNKT(Name;Befehl;[Titel;Punkt])

(Englisch: ADDMENUITEM)

Fügt einen neuen Menüpunkt Name an der Position Titel, Punkt hinzu und weist ihm den Befehl Befehl zu

Name: Ist ein Text, der den Menüpunkt-Text bestimmt.

Befehl: Ist der Befehl, der bei Auswahl dieses Menüpunktes ausgeführt werden soll. Dies kann ein beliebiger Makro- bzw. ARexxbefehl von TurboCalc sein.

Titel: Ist die Titelnummer, in der der Menüpunkt angefügt wird (0=Anfang, 1=nach erster Position...). Fehlt Titel, so wird er automatisch an das Ende gesetzt. Wird ein Leertext ("") angegeben, so ist dieser Menüpunkt nicht auswählbar!

Punkt: Ist die Menüpunktnummer, an die der Menüpunkt angefügt wird (0=Anfang, 1=nach erster Position...). Fehlt Punkt, so wird er automatisch an das Ende gesetzt.

Beispiel und weitere Hinweise siehe NEUESMENÜ

```
ADDMENUITEM("Makro 1";"MACROPLAY(Makro1)";1)
```

Dieser Befehl fügt den Menüpunkt "Makro 1" an das Ende des Bearbeiten-Menüs an. Wird der Menüpunkt ausgewählt, so wird das Makro mit dem Namen Makro1 gestartet!

```
ADDMENUITEM("~")
```

Wird als Text die Tilde ("~") angegeben, so entspricht dies einem Trennstrich. Dies kann benutzt werden, um Menügruppen voneinander zu trennen.

1.123 NEUERMENÜTITEL(Name;[Titel])

NEUERMENÜTITEL(Name;[Titel])

(Englisch: ADDMENUTITLE)

Fügt einen neuen Menütitel Name an der Position Titel ein

Name: Ist ein Text, der den Menütitel-Text bestimmt

Titel: Ist die Titelnummer, an die der Menütitel angefügt wird (0=Anfang, 1=nach erster Position...). Fehlt Titel, so wird er automatisch an das Ende gesetzt.

Beispiel und weitere Hinweise siehe NEUESMENÜ

1.124 NEUERMENÜUNTER(Name;Befehl;[Titel;Punkt;Unter])

NEUERMENÜUNTER(Name;Befehl;[Titel;Punkt;Unter])

(Englisch: ADDMENUSUB)

Fügt einen neuen Menüunterpunkt Name an der Position Titel, Punkt, Unter hinzu und weist ihm den Befehl Befehl zu

Name: Ist ein Text, der den Menüunterpunkt-Text bestimmt.

Befehl: Ist der Befehl, der bei Auswahl dieses Menüunterpunktes ausgeführt werden soll. Dies kann ein beliebiger Makro- bzw. ARexxbefehl von TurboCalc sein. Wird ein Leertext ("") angegeben, so ist dieser Menüpunkt nicht auswählbar!

Titel: Ist die Titelnummer, in der der Menüpunkt angefügt wird (0=Anfang, 1=nach erster Position...). Fehlt Titel, so wird er automatisch an das Ende gesetzt.

Punkt: Ist die Menüpunktnummer, in der der Menüunterpunkt angefügt wird (0=Anfang, 1=nach erster Position...). Fehlt Punkt, so wird er automatisch an das Ende gesetzt.

Unter: Ist die Menüunterpunktnummer, an die der Menüunterpunkt angefügt wird (0=Anfang, 1=nach erster Position...). Fehlt Unter, so wird er automatisch an das Ende gesetzt.

Beispiel und weitere Hinweise siehe NEUESMENÜ

1.125 NEUESMENÜ()

NEUESMENÜ()

(Englisch: NEWMENU)

Löscht das aktuelle Menü und erlaubt so das Erstellen eines komplett neuen Menüs mit ADDMENUTITLE, ADDMENUITEM und ADDMENUSUB

Hinweis: Normalerweise werden die Menüs nach einem NEUERMENÜ... bzw. LÖSCHEMENÜ... sofort geändert. Bei einer kompletten Menüerstellung ist dies jedoch sehr zeitaufwendig. Deswegen wird das "Erneuern" nach einem NEUESMENÜ ausgeschaltet, bis ein ZEIGEMENÜ folgt. Dies muß also der letzte Befehl einer Menüdefinition sein.

(Werden nachträglich einige Menüpunkte hinzugefügt bzw. entfernt, so ist dazu kein ZEIGEMENÜ nötig - außer es wurde zuvor NEUESMENÜ aufgerufen)

Beispiel:

NEWMENU

ADDMENUTITLE Datei

ADDMENUITEM öffnen... LOAD

ADDMENUITEM "Speichern als..." SAVEAS

ADDMENUITEM ENDE!!! QUIT

SHOWMENU

Ein weiteres Beispiel finden Sie in der Datei TurboCalc.STD2 - dort sind die Makrobefehle für das Standard-TurboCalc-Menü zu finden.

1.126 ZEIGEMENÜ()

ZEIGEMENÜ()

(Englisch: SHOWMENU)

Zeigt das geänderte Menü an, nachdem mit NEUESMENÜ das automatische Anzeigen ausgeschaltet wurde (siehe dort).

Dieser Befehl sollte also am Ende der Menüdeklaration in Skript-Dateien stehen, ansonsten kann es sein, daß das Menü nicht (vollständig) zu sehen ist!

1.127 Steuerung

Steuerung

AUSFÜHREN(Datei;Parameter;[Fenster])

BEENDEN([Flag])

BLOCKNAMENFESTLEGEN(Name;Block)

DRUCKEN(Drucker;File;NLQ;Bereich;Seite1;Seite2;LPI;Farbig;Pause;Größe;Breite;Höhe)

IFFDRUCKEN(Datei;Breite;Höhe;Tiefe;Block)

MAKROABSPIELEN(Zelle)

NAMENFESTLEGEN(Name;Wert)

NAMENLÖSCHEN(Name)

NEUBERECHNEN([Modus])

NEUETABELLE(Name)

RUN(Datei;Parameter;[Fenster])

RX(Datei;Port)

START(Dateiname)

TABELLEEINBLENDEN(Tabellenname;Fensternummer)

TABELLESCHLIEßEN(Sofort)

TABELLEVERBERGEN(Tabellenname;Fensternummer)

UNGEÄNDERT()

WÄHLETABELLE(Name[;FensterNummer])

1.128 AUSFÜHREN(Datei;Parameter;[Fenster])

AUSFÜHREN(Datei;Parameter;[Fenster])

(Englisch: EXECUTE)

Startet eine externe AmigaDOS-Batch-Datei von TurboCalc aus und wartet, bis sie wieder beendet wird.

Dieser Befehl entspricht RUN, nur daß die Datei nicht gestartet sondern ausgeführt wird. (Ab OS2.0 mit gesetztem Skript-Flag kann hierzu auch einfach RUN benutzt werden).

Einzelheiten zu den Parametern siehe deshalb bei RUN.

Beispiel:

EXECUTE("s:test")

Startet die Datei "s:test", die aus AmigaDOS-Befehlen bestehen sollte.

1.129 BEENDEN([Flag])

BEENDEN([Flag])

(Englisch: QUIT)

Dieser Befehl beendet TurboCalc.

Wird er von einem ARexx-Skript aus aufgerufen, so wird an dieses zuerst die Meldung >OK< (d.h. Fehlerwert 0) zurückgeschickt, so daß es normal beendet werden kann. Nach diesem Befehl sollte TurboCalc nicht mehr per ARexx angesprochen werden.

Flag: Bestimmt ob vor dem Beenden ein Fenster geöffnet werden soll, und zwar:

0 bzw. FALSCH: keine Abfrage (selbst falls ungesicherte Tabellen existieren)

1 bzw. WAHR: Abfrage, falls ungesicherte Tabellen existieren

Wird dieser Parameter nicht mit angegeben, so wird WAHR angenommen (wie auch beim Menüpunkt <Datei-Beenden>.

1.130 BLOCKNAMENFESTLEGEN(Name;Block)

BLOCKNAMENFESTLEGEN(Name;Block)

(Englisch: BLOCKVARIABLE)

Fügt einen "Namen" in die Namensliste ein. Der Name kann dann in Formeln benutzt werden, Einzelheiten siehe bei "Namen".

Name: Ist ein Text, der den Namen festlegt. (Achten Sie möglichst auf eine einfache Namensgebung ohne Leerzeichen oder sonstigen Sonderzeichen)

Block: Ist der Block, der dem Namen zugewiesen werden soll. Wird dieser Parameter ausgelassen, so wird der aktuelle Block benutzt.

Hinweis: Soll eine Zahl oder ein Text einem Namen zugewiesen werden, so benutzen sie hierfür NAMENFESTLEGEN.

Beispiel:

BLOCKNAMENFESTLEGEN("test";A1:C5)

weist dem Namen test den Block A1:C5 zu. Der Block kann nun unter diesem Namen angesprochen werden.

BLOCKNAMENFESTLEGEN("test")

der aktuelle Block kann nun als test angesprochen werden.

1.131 DRUCKEN(Drucker;File;NLQ;Bereich;Seite1;Seite2;LPI;Farbig;Pause;Größe;Breite;Höhe)

DRUCKEN(Drucker;File;NLQ;Bereich;Seite1;Seite2;LPI;Farbig;Pause;Größe;Breite;Höhe)

(Englisch: PRINT)

Drucker: Bestimmt, ob die Ausgabe zum Drucker (WAHR bzw. ungleich Null) oder aber in eine Datei gehen soll (bei 0 bzw. FALSCH)

File: Ist ein Text, der den Dateinamen für die Druckausgabe in eine Datei angibt.

NLQ: Bestimmt die Qualität: WAHR bzw. ungleich Null ist hoch, bei 0 bzw. FALSCH wird die Tabelle als Entwurf gedruckt.

Bereich: Bei WAHR (ungleich Null) wird alles gedruckt, bei FALSCH (0) nur die folgenden Seiten:

Seite1, Seite2: Geben die Seiten an, die gedruckt werden sollen.

LPI: 6 (WAHR bzw ungleich Null) oder 8 (FALSCH bzw. 0) lpi (lpi = Zeilen pro Zoll).

Farbig: Druckt farbig (WAHR bzw. ungleich Null) oder schwarz/weiß (FALSCH bzw. 0).

Pause: Druckt alles am Stück (WAHR bzw. ungleich Null) oder macht nach jeder Seite eine Pause (FALSCH bzw. 0).

Größe: Bestimmt die Papiergröße

0: A4 normal (21*29.7)

1: A4 endlos (21*30.5)

2: A3 normal (29.7*42)

3: Benutzerdefiniert:

Breite, Höhe: Bestimmt die Papierbreite und -höhe für "benutzerdefiniert", beide Angaben in Zentimeter..

Ausgelassene Parameter werden nicht geändert. Werden keine Parameter angegeben, so entspricht dies <Datei-Drucken>, es erscheint dann ein entsprechendes Fenster zur Auswahl der Optionen.

Weitere Druckparameter können mit DRUCKOPTIONEN eingestellt werden.

Tip: Siehe auch IFFDRUCKEN.

Beispiel:

DRUCKEN(;

Druckt die aktuelle Tabelle wie zuvor aus, es werden keine Parameter geändert

DRUCKEN()

zeigt das Fenster zur Druckauswahl an (entspricht <Datei-Drucken...>)

DRUCKEN(0;"ram:print.out")

Druckt die aktuelle Tabelle in die Datei "ram:print.out" - die restlichen Parameter bleiben unverändert.

DRUCKEN(0;"ram:print.out";;0;2;3)

dito, jedoch nur die Seiten 2 und 3

1.132 IFFDRUCKEN(Datei;Breite;Höhe;Tiefe;Block)

IFFDRUCKEN(Datei;Breite;Höhe;Tiefe;Block)

(Englisch: IFFPRINT)

Damit wird ein Tabellenteil als Iff-Bild gespeichert. Dies ermöglicht einen (automatischen) Export zu anderen Programmen.

Datei: Ist der Name, unter dem das Bild abgespeichert werden soll. Wird er ausgelassen, so wird der zuletzt angegebene benutzt (entweder bei diesem Befehl, bei DRUCKEN oder im Drucken-Requester). Wird "" angegeben, so wird nach einem Dateinamen gefragt (mittels

Dateirequester). (Anmerkung: Hierbei wird die Makroausführung angehalten, bis der Name angegeben wurde bzw. auf Abbruch gedrückt wurde)

Breite/Höhe/Tiefe: Das Bild-Format. Wird es ausgelassen, so bleibt die alte Einstellung erhalten

Block: Bestimmt, welcher Block als Bild gespeichert werden soll. Wird dieser Parameter ausgelassen, so wird der aktuell ausgewählte Block benutzt.

(Anmerkung: Dies stimmt normalerweise nicht mit dem Druckbereich überein! Soll dies geschehen, so kann als Block DRUCKEBREICH angegeben werden, siehe letztes Beispiel)

Beispiel:

```
IFFDRUCKEN("ram:test";640;512;3;A1:D20)
```

Speichert den Bereich A1:D20 als Bild mit dem Namen ram:test und der Größe 640*512*3 (3=8 Farben)

```
IFFDRUCKEN("ram:test";;;;DRUCKEBREICH)
```

Speichert den aktuellen Druckbereich unter ram:test mit der Standardgröße (d.h. die zuletzt angegebene Größe). Wurde kein Druckbereich bestimmt, d.h. die Variable DRUCKEBREICH ist nicht definiert, so erscheint eine Fehlermeldung.

1.133 MAKROABSPIELEN(Zelle)

MAKROABSPIELEN(Zelle)

(Englisch: MAKROPLAY)

Startet das Makro ab der Zelle Zelle. Fehlt diese Angabe, so entspricht dieser Befehl <Makro-Abspielen> und öffnet ein Fenster, in dem das Makro ausgewählt werden kann.

Zelle: Ist eine beliebige Zellangabe (auch Namen möglich) und kann auch auf andere Tabellen verweisen, indem die AT-Funktion (oder @) angewendet wird.

Wird der Befehl per ARexx ausgeführt, so wartet ARexx bis das Makro beendet wurde.

Achtung: Dieser Befehl ist nicht dazu gedacht, innerhalb eines Makros, Unterroutrinen aufzurufen. Benutzen Sie dafür den Befehl ROUTINE!

1.134 NAMENFESTLEGEN(Name;Wert)

NAMENFESTLEGEN(Name;Wert)

(Englisch: VARIABLE)

Fügt einen "Namen" in die Namensliste ein. Der Name kann dann in Formeln benutzt werden, Einzelheiten siehe "Namen"

Name: Ist ein Text, der den Namen festlegt. (Achten Sie möglichst auf eine einfache Namensgebung ohne Leerzeichen oder sonstigen Sonderzeichen)

Wert: Ist der Wert, der dem Namen zugewiesen werden soll. Es kann sich dabei um eine Zahl, einen Wahrheitswert oder einen Text handeln.

Texte können auch mit einem Gleichheitszeichen "=" beginnen, und so eine Formel bzw. einen Bezug/Bereich darstellen, siehe Beispiele.

Hinweis: Soll ein Block als Name festgelegt werden, so kann dies entweder über einen Text ("=A1:C5") geschehen oder aber den Befehl BLOCKNAMENFESTLEGEN dafür benutzen.

Werden keine Parameter angegeben, so entspricht dies dem Menüpunkt <Befehle-Namen festlegen...>. Es erscheint dann ein entsprechendes Fenster.

Beispiel:

```
NAMENFESTLEGEN("MWST";0.15)
```

Weist dem Namen MWST den Wert 0,15 zu. Nun kann in Formeln dieses als Konstante benutzt werden, etwa =200*(1+MWST). Bei einer Änderung von MWST und <Befehle-Neuberechnen> wird auch die Formel geändert

```
NAMENFESTLEGEN("Zelle1";"=C5")
```

Nun kann die Zelle C5 immer mit "Zelle1" angesprochen werden, auch in Formeln.

```
NAMENFESTLEGEN("Test";"=ZELLE(-1;-1)*2")
```

Weist dem Namen Test eine Formel zu (hier: nimmt die Zelle links darüber und verdoppelt sie).

1.135 NAMENLÖSCHEN(Name)

```
NAMENLÖSCHEN(Name)
```

(Englisch: DELETEVARIABLE)

Löscht den mit Namen bestimmten Namen aus der Namensliste. Einzelheiten siehe bei NAMENFESTLEGEN und "Namen".

Name: Ist ein Text, der den Namen angibt.

Beispiel:

```
NAMENLÖSCHEN("MWST")
```

Löscht den Namen MWST.

1.136 NEUBERECHNEN([Modus])

```
NEUBERECHNEN([Modus])
```

(Englisch: RECALC)

Ohne Parameter wird die aktuelle Tabelle neu berechnet. (entspricht <Befehle-Neuberechnen>). Dies ist sinnvoll, wenn die automatische Berechnung ausgeschaltet wurde:

Wird Modus angegeben, so kann die automatische Berechnung eingeschaltet werden (WAHR bzw. ungleich Null). Bei 0 bzw. FALSCH wird Sie ausgeschaltet.

Hinweis: Die automatische Berechnung kann auch mit "TABELLENOPTIONEN" ein- bzw. ausgeschaltet werden (dort jedoch mit weiteren Parametern) - da das Ausschalten des Berechnens jedoch oft benutzt wird (etwa bei Makros), wurde dies hiermit vereinfacht.

Beispiel:

```
NEUBERECHNEN
```

Berechnet die Formeln neu und aktualisiert die Tabelle

```
RECALC(0)
```

Schaltet die automatische Berechnung aus.

1.137 NEUETABELLE(Name)

NEUETABELLE(Name)

(Englisch: NEWSHEET)

Öffnet eine neue Tabelle und dazugehöriges Fenster. Die Tabelle erhält den mit Name angegebenen Namen bzw. einen Standardnamen, falls der Parameter fehlt.

Name: Ein Text, der den Namen der neuen Tabelle darstellt. (Eine ggf. auf Diskette oder Festplatte vorhandene Tabelle wird nicht geladen, dazu LADEN benutzen). Fehlt der Parameter, so erhält die Tabelle den Namen "Tabelle1" (durchnummeriert) und beim ersten <Datei-Speichern> wird nach dem Namen gefragt.

1.138 RUN(Datei;Parameter;[Fenster])

RUN(Datei;Parameter;[Fenster])

Startet ein externes Programm von TurboCalc aus und wartet, bis es wieder beendet wird.

Datei: Ist der Name der Datei, die gestartet werden soll. Möglichst mit Pfad angeben bzw. den Pfad so setzen, daß die Datei auch gefunden wird.

Parameter: Ist der Parametertext, der dem Programm übergeben werden soll. Sind es mehrere Parameter, so einfach (wie üblich) durch Leerzeichen trennen und in einen Text schreiben.

Fenster: Gibt den Namen des Fensters an, das vor Starten des Programms geöffnet werden soll. Fehlt diese Angabe, so wird "CON:0/0/640/200/TurboCalc/CLOSE/WAIT/AUTO" angenommen. Ab OS2.0 wird dadurch nur ein Fenster geöffnet, wenn es wirklich benötigt wird. Weiterhin bleibt dann das Fenster erhalten, bis das Schließsymbol angeklickt wird.

(Für 1.3-Benutzer erscheint nach Ausführung des Befehls die Meldung "Bitte Return drücken" und es wird gewartet, bis diese Taste gedrückt wird.)

Achten Sie darauf, daß die Fensterangabe korrekt ist, ansonsten ist ein Ausführen des Befehls nicht möglich.

Beispiel:

RUN("dir";"df0:") zeigt das Verzeichnis des Laufwerks df0: an.

Ausdrucken im Hintergrund:

=PRINT(0;"ram:drucker.out")

=RUN("RUN";">nil: copy ram:drucker.out prt: QUIET")

Dieses Makro druckt in die Datei "ram:drucker.out" und startet dann über den Befehl "RUN" (so daß es im Hintergrund läuft) den Befehl "copy", der die Datei zum Drucker schickt. So kann während des Ausdrucks weitergearbeitet werden.

1.139 RX(Datei;Port)

RX(Datei;Port)

Mit diesem Befehl können externe REXX-Scripts gestartet werden.

Datei ist ein Text mit dem Dateinamen.

Port: Bestimmt den Port, an den 'Datei' geschickt werden soll. Wird er ausgelassen, so wird dafür "REXX" verwandt und 'Datei' an den ARexx-Port geschickt. Damit können Befehle, die für andere Anwendungen bestimmt sind, direkt dorthin weitergeleitet werden.

Der Rückgabewert des Scripts (Ok/Fehler/Warnung) wird (falls dieser Befehl in einem Makro verwandt wird, in die entsprechende Zelle geschrieben, in der der Makrobefehl steht).

Anmerkung: Wird Port angegeben, so sollte 'Datei' eher 'Befehl' lauten, da dies dann meist direkt den Befehl angibt (im Gegensatz zur Anwendung ohne Port, in dem dies eine Datei spezifizierte, sofern der Befehl nicht in '.' gesetzt wurde)

Beispiel:

`RX("Beispiel.rexx")` - führt die Datei `Beispiel.rexx` aus

`RX("Beispiel.rexx";"REXX")` - dito

`RX("befehl";"PORT")` - schickt an den ARexx-Port `PORT` 'befehl'

`RX("'address PORT befehl'")` - dito, jedoch über den Umweg `REXX`.

1.140 START(Dateiname)

`START(Dateiname)`

Dieser Befehl ruft eine externe TurboCalc-Befehlsdatei auf. Diese besteht aus zeilenweise geschriebenen Makrobefehlen (ohne Anführungszeichen). (Die Befehle können dabei sowohl im Makro-Stil mit Klammern oder im ARexx-Stil durch Leerzeichen getrennt sein)

Die meisten Makrospezifischen Befehle wie etwa `SPRINGE`, `SPRINGEWENN`, `MWENN`, `ROUTINE` oder `FÜR` sind nicht verwendbar!

Dies ist eine optionale Methode, Makro-Programme zu schreiben. Sie eignet sich für längere Programme (hat jedoch das Manko, daß Verzweigungen sowie Schleifenbefehle nicht möglich sind)

Hinweis: Um Fehler besser lokalisieren zu können, gibt TurboCalc bei einem aufgetretenen Fehler die Fehlerzelle an. In diesem Fall kann die Spalte (der Buchstabe) ignoriert werden, die Zeile entspricht der Zeile der Datei (Beispiel: `A100`: Fehler in Zeile 100).

Tip: Als Dateiname kann auch `$$` angegeben werden, dann wird die interne Skript-Datei (die zu Beginn ausgeführt wird) gestartet. Sie stellt die Menüs und den Bildschirm ein. Diese Datei ist unter `TurboCalc.std2` beigefügt und kann betrachtet und geändert werden, siehe im Kapitel "Makro" bei "Makro beim Start von TurboCalc ausführen"

Beispiel:

`START("ram:test")`

und z.B. folgender Datei `ram:test`:

`SELECT a1`

`PUT 3`

`SELECT A1:A10`

`REIHEBERECHNEN(1;3)`

1.141 TABELLEEINBLENDEN(Tabellenname;Fensternummer)

`TABELLEEINBLENDEN(Tabellenname;Fensternummer)`

(Englisch: `SHEETSHOW`)

Blendet das angegebene Fenster wieder ein (falls es mit `TABELLEVERBERGEN` ausgeblendet wurde) bzw. bringt es in den Vordergrund und aktiviert es.

Name: Dies gibt den Namen der Tabelle an (d.h. den Namen, der als Fenstertitel zu sehen ist, jedoch ohne "-2..."). Falls die Endung `".TCD"` ist, so kann sie auch ausgelassen werden. Weiterhin können Pfadangaben weggelassen werden.

Für die Datei `"DH0:TurboCalc/Tabellen/Beispiel.TCD"` sind also folgende Namen korrekt: `"Beispiel"`, `"Beispiel.TCD"`, `"TurboCalc/Beispiel"...`

FensterNummer: Gibt die Nummer des Fensters an (d.h. den "-XXX"-Teil des Fenstertitels). Die Numerierung beginnt bei 1. Wird der Parameter ausgelassen bzw. ist er zu groß, so wird das erste Fenster dieser Tabelle genommen.

Achtung: Dieses Fenster wird dadurch nicht automatisch zum "aktuellen" Fenster bei der Makroausführung. Dazu WÄHLETABELLE benutzen!

Im Makromodus: Wird dieser Befehl im Makromodus benutzt, so wird der Zelle, in der dieser Befehl steht, der Wert WAHR zugewiesen, falls die Tabelle gefunden wurde. Konnte die Tabelle nicht gefunden werden, so enthält diese Zelle den Wert FALSCH. Dies kann dann mit SPRINGEWENN abgefragt werden (siehe bei "Rückgabewerte von Befehlen" am Beginn der Befehlsübersicht).

Wird kein Parameter angegeben, so entspricht dies <Optionen-Fenster einblenden>, es erscheint dann ein Auswahlfenster.

1.142 TABELLESCHLIEßEN(Sofort)

TABELLESCHLIEßEN(Sofort)

(Englisch: CLOSESHEET)

Schließt die aktuelle Tabelle (ggf. mit Fragen, ob die Änderungen gespeichert werden sollen). War es die letzte offene Tabelle, so wird TurboCalc beendet.

Sofort: Bestimmt, ob bei Änderungen gefragt wird, ob diese gespeichert werden sollen.

0: keine Abfrage

ansonsten wird abgefragt (also 1 oder Auslassen des Parameters)

Hinweis: Achten Sie darauf, daß nach dem Schließen einer Tabelle jede andere die aktive sein kann. Wenn Sie weitere Befehle eingeben möchten, verwenden Sie bitte als nächstes den Befehl WÄHLETABELLE..

1.143 TABELLEVERBERGEN(Tabellenname;Fensternummer)

TABELLEVERBERGEN(Tabellenname;Fensternummer)

(Englisch: SHEETHIDE)

Blendet das angegebene bzw. das aktuelle Fenster aus. Ist nur noch ein Fenster bei TurboCalc geöffnet, so wird eine Fehlermeldung ausgegeben. Ein Fenster muß immer geöffnet sein, ggf. vorher eine neue Tabelle öffnen bzw. erzeugen!

Name: Dies gibt den Namen der Tabelle an (d.h. den Namen, der als Fenstertitel zu sehen ist, jedoch ohne "-2..."). Falls die Endung ".TCD" ist, so kann sie auch ausgelassen werden. Weiterhin können Pfadangaben weggelassen werden.

Für die Datei "DH0:TurboCalc/Tabellen/Beispiel.TCD" sind also folgende Namen korrekt: "Beispiel", "Beispiel.TCD", "TurboCalc/Beispiel"...

FensterNummer: Gibt die Nummer des Fensters an (d.h. den "-XXX"-Teil des Fenstertitels). Die Numerierung beginnt bei 1. Wird der Parameter ausgelassen bzw. ist er zu groß, so wird das erste Fenster dieser Tabelle genommen.

Wird kein Parameter angegeben, so wird das aktuelle Fenster verborgen.

Im Makromodus: Wird dieser Befehl im Makromodus benutzt, so wird der Zelle, in der dieser Befehl steht, der Wert WAHR zugewiesen, falls die Tabelle gefunden wurde. Konnte die Tabelle nicht gefunden werden, so enthält diese Zelle den Wert FALSCH. Dies kann dann mit SPRINGEWENN abgefragt werden (siehe bei "Rückgabewerte von Befehlen" am Beginn der Befehlsübersicht).

Tip: Diese Funktion kann gut für spezielle Makros eingesetzt werden: Das Makro fügt einen Menüpunkt hinzu und verbirgt sich dann selbst. Dadurch stört die Makrotabelle nicht auf dem Bildschirm, die Funktion kann jedoch bequem per Menü (bzw. auch Makrorequester) aufgerufen werden.

1.144 UNGEÄNDERT()

UNGEÄNDERT()

(Englisch: UNCHANGED)

Dieser Befehl markiert die aktuelle Tabelle als "nicht geändert", d.h. beim Schließen der Tabelle (bzw. Beenden von TurboCalc) erscheint keine Abfrage, ob die Tabelle gespeichert werden soll.

Tip: Dies kann für Makros benutzt werden, die ihre Tabelle (etwa zum Speichern von Zwischenergebnissen) modifizieren. So kann vermieden werden, daß beim Beenden von TurboCalc nachgefragt wird, ob das Makro gespeichert werden soll.

1.145 WÄHLETABELLE(Name[;FensterNummer])

WÄHLETABELLE(Name[;FensterNummer])

(Englisch: SELECTSHEET)

Dieser Befehl wählt eine Tabelle als neue Makro/Arexx-Tabelle aus, d.h. die Tabelle auf die sich alle Befehle und Zellverweise beziehen.

Dieser Befehl aktiviert jedoch nicht automatisch ein bestimmtes Fenster, verwenden Sie dazu den Befehl FENSTERAKTIVIEREN bzw. FENSTEREINBLENDEN!

Name: Dies gibt den Namen der Tabelle an (d.h. den Namen, der als Fenstertitel zu sehen ist, jedoch ohne "-2..."). Falls die Endung ".TCD" ist, so kann sie auch ausgelassen werden. Weiterhin können Pfadangaben weggelassen werden.

Für die Datei "DH0:TurboCalc/Tabellen/Beispiel.TCD" sind also folgende Namen korrekt: "Beispiel", "Beispiel.TCD", "TurboCalc/Beispiel"...

FensterNummer: Gibt die Nummer des Fensters an (d.h. den "-XXX"-Teil des Fenstertitels). Die Numerierung beginnt bei 1. Wird der Parameter ausgelassen bzw. ist er zu groß, so wird das erste Fenster dieser Tabelle genommen.

Im Makromodus: Wird dieser Befehl im Makromodus benutzt, so wird der Zelle, in der dieser Befehl steht, der Wert WAHR zugewiesen, falls die Tabelle gefunden wurde. Konnte die Tabelle nicht gefunden werden, so enthält diese Zelle den Wert FALSCH. Dies kann dann mit SPRINGEWENN abgefragt werden (siehe bei "Rückgabewerte von Befehlen" am Beginn der Befehlsübersicht).

Tip: Möchte man den Fensternamen zu einem bestimmten Zeitpunkt auslesen, so kann man dazu die Funktion TABELLEN-NAME benutzen. (Etwa um das aktuelle Fenster zwischenspeichern).

Beispiel:

WÄHLETABELLE("Tabelle1")

wählt die Tabelle aus, die zu Beginn normalerweise geöffnet wird.

SELECTSHEET("Tabelle1";2)

dito, jedoch nun ein ggf. mit <Optionen-Fenster teilen> erzeugtes zweites Tabellenfenster.

1.146 Bildschirm

Bildschirm

ALTEFARBEN()

BILDSCHIRM(Breite;Höhe;Tiefe;Modus)

FARBEÄNDERN(Farbe;Rot;Grün;Blau)

FENSTERAKTIVIEREN()

FENSTERANORDNEN()

FENSTERANORDNEN2()
FENSTERGRÖßE(Breite;Höhe)
FENSTERNACHHINTEN()
FENSTERNACHVORN()
FENSTERPOSITIONIEREN(X;Y;Breite;Höhe)
FENSTERVERSCHIEBEN(X;Y)
IKONIFIZIEREN()
STDFARBEN()
ZEICHENSATZ(Zeichensatz;Modus)

1.147 ALTEFARBEN()

ALTEFARBEN()

(Englisch: OLDCOLORS)

Setzt die Farben so, wie sie beim Starten von TurboCalc vorgefunden wurden. Dies ist praktisch, wenn man in einem Makro (vorübergehende) Farbmanipulationen durchführen möchte und am Ende die alte Einstellung wieder setzen möchte.

1.148 BILDSCHIRM(Breite;Höhe;Tiefe;Modus)

BILDSCHIRM(Breite;Höhe;Tiefe;Modus)

(Englisch: SCREEN)

Bestimmt, ob TurboCalc einen eigenen Bildschirm öffnen soll oder ob die Fenster auf dem WorkBench-Bildschirm erscheinen sollen. Weiterhin bestimmt dieser Befehl auch das Aussehen des Bildschirms:

Breite: Bestimmt die Breite des Bildschirms (mindestens 600!). -1 oder weglassen des Parameters entspricht der Breite des WorkBench-Bildschirm. 0 bedeutet: Kein eigener Bildschirm, benutze WorkBench-Bildschirm (die restlichen Parameter spielen dann keine Rolle mehr!)

Höhe: Bestimmt die Höhe des Bildschirms (mindestens 200!). -1 oder weglassen des Parameters entspricht der Höhe des WorkBench-Bildschirm.

Tiefe: Bestimmt die Anzahl der Bitplanes, d.h. die Anzahl der Farben. -1 oder weglassen des Parameters übernimmt die Farbanzahl von der WorkBench. 1 entspricht 2 Farben, 2 = 4 Farben, 3 = 8 Farben...

Modus: Bestimmt den Bildschirmmodus, mit dem der Bildschirm geöffnet werden soll (erst ab OS2.0 möglich). Dieser Parameter wird nur für "ungewöhnliche" Formate benötigt. Bei den normalen wird dies anhand der Bildschirmausmaße erkannt.

Dieser Modus kann entweder als Zahl (die Bildschirm-ID) oder als Buchstabenkombination eingegeben werden.

Wird kein Parameter angegeben, so wird ein Fenster geöffnet, in dem der Benutzer den Bildschirmmodus einstellen kann. Dies entspricht dann <Optionen-Bildschirm-einstellen>.

Beispiel:

BILDSCHIRM() öffnet ein Fenster, um den Bildschirm zu bestimmen.

BILDSCHIRM(;) öffnet einen Bildschirm mit den Ausmaßen und der Farbanzahl der WorkBench.

BILDSCHIRM(0) schließt einen ggf. geöffneten Bildschirm und setzt alle Fenster auf die WorkBench

BILDSCHIRM(640;256;3) öffnet ein 640 Punkte breiten, 256 Punkte hohen Bildschirm mit 8 Farben.

SCREEN(640;470;2;"VP") öffnet einen 640*480 VGA-Productivity-Bildschirm

1.149 FARBEÄNDERN(Farbe;Rot;Grün;Blau)

FARBEÄNDERN(Farbe;Rot;Grün;Blau)

(Englisch: CHANGECOLOR)

Hiermit kann eine Farbe des TurboCalc-Bildschirms (bzw. des WorkBench-Bildschirms, falls sich die TurboCalc-Fenster auf diesem Bildschirm befinden) eingestellt werden:

Farbe: dies bestimmt die Farbnummer, die eingestellt werden soll. (0=Hintergrundfarbe, 1=Farbe 1, usw.)

Rot: Bestimmt den Rotanteil der Farbe von 0 bis 15. (0=kein Rotanteil, 15=voller Rotanteil)

Grün: Bestimmt den Grünanteil der Farbe von 0 bis 15. (0=kein Grünanteil, 15=voller Grünanteil)

Blau: Bestimmt den Blauanteil der Farbe von 0 bis 15. (0=kein Blauanteil, 15=voller Blauanteil)

Wird kein Parameter angegeben, so erscheint ein Requester zur Farbeinstellung (siehe Menüpunkt "Optionen-Bildschirm-Farben").

Tip: Ist man sich über die Parameter nicht ganz sicher, so kann man den Farbrequester (<Optionen-Bildschirm-Farben>) dazu benutzen. (Die Farbnummer der darin enthaltenen Farben beginnt bei 1, es muß also 1 subtrahiert werden - die Rot/Grün/Blau-Anteile können übernommen werden).

Beispiel:

FARBEÄNDERN(0;0;0;0)

setzt die Hintergrundfarbe auf Schwarz

CHANGECOLOR(0;15;15;15)

und auf Weiß.

CHANGECOLOR(1;15;0;0)

Schreibfarbe (Farbe 1) auf "Rot" setzen (bei der normalen Farbpalette).

1.150 FENSTERAKTIVIEREN()

FENSTERAKTIVIEREN()

(Englisch: ACTIVATEWINDOW)

Aktiviert das aktuelle Fenster.

1.151 FENSTERANORDNEN()

FENSTERANORDNEN()

(Englisch: POSWINDOW)

Entspricht <Optionen-Fenster anordnen- nebeneinander>.

1.152 FENSTERANORDNEN2()

FENSTERANORDNEN2()

(Englisch: POSWINDOW2)

Entspricht <Optionen-Fenster anordnen-überlappend>.

1.153 FENSTERGRÖßE(Breite;Höhe)

FENSTERGRÖßE(Breite;Höhe)

(Englisch: SIZEWINDOW)

Ändert die Größe des aktuellen Fensters auf dem Bildschirm:

Breite, Höhe: Bestimmt die neuen Ausmaße des Fensters

1.154 FENSTERNACHHINTEN()

FENSTERNACHHINTEN()

(Englisch: WINDOWTOBACK)

Verschiebt das aktuelle Fenster nach hinten auf dem Bildschirm.

1.155 FENSTERNACHVORN()

FENSTERNACHVORN()

(Englisch: WINDOWTOFRONT)

Verschiebt das aktuelle Fenster nach vorne auf dem Bildschirm.

1.156 FENSTERPOSITIONIEREN(X;Y;Breite;Höhe)

FENSTERPOSITIONIEREN(X;Y;Breite;Höhe)

(Englisch: CHANGEWINDOW)

Verschiebt das aktuelle Fenster auf dem Bildschirm und ändert die Größe:

X, Y: Geben die neuen Koordinaten an

Breite, Höhe: Bestimmt die neuen Ausmaße des Fensters

Dies entspricht FENSTERVERSCHIEBEN und FENSTERGRÖßE.

1.157 FENSTERVERSCHIEBEN(X;Y)

FENSTERVERSCHIEBEN(X;Y)

(Englisch: MOVEWINDOW)

Verschiebt das aktuelle Fenster auf dem Bildschirm:

X, Y: Geben die neuen Koordinaten an

1.158 IKONIFIZIEREN()

IKONIFIZIEREN()

(Englisch: ICONIFY)

Dies entspricht <Datei-Ikonifizieren> und veranlaßt, daß TurboCalc in einen "Wartemodus" geht. Es werden alle Fenster geschlossen und nur ein kleines "Erinnerungsfenster" bzw. ab OS2.0 ein Icon bleibt erhalten. Durch Klick darauf wird TurboCalc wieder so hergestellt, wie es ikonifiziert wurde.

1.159 STDFARBEN()

STDFARBEN()

(Englisch: STDCOLORS)

Setzt die Standard-Farben von TurboCalc. Dies ist nützlich, falls die Farben zuvor verstellt wurden oder aber für 1.3-Benutzer. Einzelheiten siehe beim Menüpunkt <Optionen-Bildschirm-Std-Farben>.

1.160 ZEICHENSATZ(Zeichensatz;Modus)

ZEICHENSATZ(Zeichensatz;Modus)

(Englisch: SETFONT)

Hiermit kann der globale Zeichensatz (entsprechend <Optionen-Bildschirm-Zeichensatz>) gesetzt werden, und zwar:

Zeichensatz: Ist ein Zeichensatztext (Name/Größe, etwa "Times/13"). Wird dieser Parameter ausgelassen, so erscheint ein Auswahlfenster, um den Zeichensatz auszuwählen.

Modus: Dieser Parameter bestimmt, für welchen Bereich der Zeichensatz geändert werden soll.

0 (oder auslassen): Zeichensatz für alle Menüs und alle Fenster.

1: Zeichensatz für das aktuelle Fenster

2: Menüzeichensatz

3: Standardzeichensatz der aktuellen Tabelle (entspricht somit STDSCHRIFT)

Hinweis: Per TurboCalc-Menüs kann nur der Modus 0 eingestellt werden. Unterschiedliche Zeichensätze für verschiedene Fenster können per Menü nicht ausgewählt werden. Siehe dazu auch Beispiele.

1. Beispiel:

```
=ADDMENUITEM("Menüfont";"SETFONT("";2)";5)
```

Dieser Makrobefehl fügt den Menüpunkt "Menüfont" ans Ende des Menüs "Optionen" hinzu. Wird dieser Menüpunkt ausgewählt, so kann ein neuer Zeichensatz für die Menüs bestimmt werden.

2. Beispiel

```
=ADDMENUITEM FensterFont "SETFONT # 1"
```

(in der TurboCalc.STD - Start-Datei) fügt den Menüpunkt FensterFont zur Auswahl eines neues Fensterzeichensatzes hinzu.

1.161 Sonstige Befehle

Sonstige Befehle

AUFNAHME()

COMMAND()

DIAEINBLENDEN()

DRUCKVORSCHAU()

EDIT()

FORMELEINFÜGEN()

MAKROEINFÜGEN()

NAMENEINFÜGEN()

NEUESFENSTER()
OBJEKT(Name;Typ;Text;Wert1;Wert2;Block)
OBJEKTAUSWAHL(Name)
OBJEKTKLASSE(Name)
OBJEKTPARA(Name;Makro;Makroaktivierung;Farbe;Rahmen)
OBJEKTPOS(Name;X;Y;Breite;Höhe)
RÜCKGÄNGIG()
SPEZIALOPTIONEN(Flags)
STOPAUFNAHME()
SYSINFO()
TCMAKRO(TCLib;Offset;Num1;Num2;Text)
ÜBER()
WIEDERHERSTELLEN()

1.162 AUFNAHME()

AUFNAHME()

(Englisch: RECORD)

Entspricht <Makro-Aufnahme starten> - Dieser Befehl sollte nicht in einem Makro verwendet werden, sondern dient nur zur Menüerstellung (also nur hinter NEUERMENÜPUNKT angeben!).

1.163 COMMAND()

COMMAND()

Dieser Befehl entspricht also einem Druck auf <F10> (bei der Standard-Tastenbelegung). Dieser Befehl hat im normalen Makro-Modus keinen großen Sinn, er wurde hinzugefügt, um diese Funktion mit einer Tastenkombination zu verbinden oder aber einen entsprechenden Menüpunkt zu definieren.

Beispiel:

```
KEY("F1";"COMMAND")
```

1.164 DIAEINBLENDEN()

DIAEINBLENDEN()

(Englisch: DIASHOW)

Entspricht dem Menüpunkt <Daten-Diagramm einblenden...>

1.165 DRUCKVORSCHAU()

DRUCKVORSCHAU()

(Englisch: PREVIEW)

Ruft die Druckvorschau auf, entspricht <Datei-Druckvorschau>.

1.166 EDIT()

EDIT()

Dieser Befehl aktiviert die Eingabezeile, entspricht also einem Druck auf <F2> (bei der Standard-Tastenbelegung). Dieser Befehl hat im normalen Makromodus keinen großen Sinn, er wurde hinzugefügt, um diese Funktion mit einer Tastenkombination zu verbinden oder aber einen entsprechenden Menüpunkt zu definieren.

Beispiel:

```
KEY("F1";"EDIT")
```

1.167 FORMELEINFÜGEN()

FORMELEINFÜGEN()

(Englisch: INSERTFORMULA)

Entspricht dem Menüpunkt <Befehle-Einfügen-Formel...>, öffnet also ein Fenster zur Auswahl einer Formelfunktion, die in die aktuelle Zelle eingefügt werden soll.

1.168 MAKROEINFÜGEN()

MAKROEINFÜGEN()

(Englisch: INSERTMACRO)

Entspricht dem Menüpunkt <Befehle-Einfügen-Makro...>, öffnet also ein Fenster zur Auswahl eines Makrobefehls, der in die aktuelle Zelle eingefügt werden soll.

1.169 NAMENEINFÜGEN()

NAMENEINFÜGEN()

(Englisch: INSERTNAME)

Entspricht dem Menüpunkt <Befehle-Einfügen-Namen...>, öffnet also ein Fenster zur Auswahl eines Namens, der in die aktuelle Zelle eingefügt werden soll.

1.170 NEUESFENSTER()

NEUESFENSTER()

(Englisch: NEWWINDW)

Entspricht <Optionen-Fenster teilen>.

1.171 OBJEKT(Name;Typ;Text;Wert1;Wert2;Block)

OBJEKT(Name;Typ;Text;Wert1;Wert2;Block)

(Englisch: OBJECT)

Erstellt ein neues Objekt, siehe <Daten-Objekt erstellen>. Wird kein Parameter angegeben, so wird auch dieser Menüpunkt aufgerufen.

Name: Der Name des Objekts. Der Name kann für OBJEKTAUSWAHL sowie OBJEKTPARA zur Auswahl benutzt werden.

Typ: Der Objekt-Typ (Text, Bild, Grafik...) (ab 0 durchnummeriert, siehe auch Anmerkung unten)

Text: Text, der als Parameter dem Objekt übergeben wird, siehe unten

Wert1, Wert2: Zwei Werte, die als Parameter dem Objekt übergeben werden

Block: Bestimmt die Position/Größe des Objekts.

Objekt-Parameter: Text, Wert1 und Wert2 werden dem Objekt übergeben und haben, je nach Objekt-Typ, unterschiedliche Bedeutung. Somit ist es möglich, bei der Objekterstellung per Makro, das Aussehen anzupassen:

Textobjekt: Text: ist der eigentliche Text, Wert1/2: Formatierung

Zeichnungsobjekt: Wert1: Typ, Wert2: Formatierung

Bildobjekt: Text: Bildname

Nichtbenutzte Parameter können ausgelassen werden bzw. durch beliebige Werte ("" bzw. 0 sind empfehlenswert!) ersetzt werden.

Anmerkung: Der Parameter Typ ist entweder die Nummer der Objektklasse oder aber kann wie folgt bestimmt werden: Angabe als ObjektID, die mittels der Funktion OBJID aus dem entsprechenden Objekttyp-Kürzel ermittelt werden kann. Diese Variante ist dem 'durchzählen' vorzuziehen, da sie unabhängig von der Reihenfolge ist.

Die beiliegenden Objektklassen haben folgende Kürzel:

TEXT Textklasse

GEL Grafische Elemente

IFF IFF-Bilder

CHRT Diagramme

LOGO externe Beispielklasse

Beispiel:

```
OBJEKT("Test";OBJID("TEXT");"Dies ist ein Test-Textobjekt")
```

erstellt ein Textobject an der aktuellen Cursorposition

1.172 OBJEKTAUSWAHL(Name)

OBJEKTAUSWAHL(Name)

(Englisch: OBJECTSELECT)

Wählt das Objekt Name aus. (Entspricht einem Klick mit der Maus auf das Objekt)

Name: Name des Objekts.

1.173 OBJEKTKLASSE(Name)

OBJEKTKLASSE(Name)

(Engl: OBJECTCLASS)

Hiermit können als TCLib vorliegende Objektklassen, die beim Start nicht automatisch eingebunden wurden (etwa weil sie nicht in einem der Verzeichnisse LIBS:TCLibs oder PROGDIR:TCLibs liegen) nachträglich angemeldet werden.

Name: Ist dabei der Name der Objektklasse, ggf. mit Pfad.

Beispiel:

```
OBJEKTKLASSE("dh0:TurboCalc/object_extra.tclib")
```

1.174 OBJEKTPARA(Name;Makro;Makroaktivierung;Farbe;Rahmen)

OBJEKTPARA(Name;Makro;Makroaktivierung;Farbe;Rahmen)

(Englisch: OBJECTPARA)

Bestimmt die allgemeinen Parameter für ein Objekt.

Name: Ist der Name des Objekts. Wird er ausgelassen, so wird das aktuell ausgewählte Objekt verwandt. Ist kein Objekt ausgewählt, so erscheint eine Fehlermeldung.

Makro: Der Makrotext, der bei Mausklick ausgeführt werden soll.

Makroaktivierung: Bestimmt, ob der Makrotext bei Mausklick ausgeführt werden soll. (0/FALSCH: Makro wird nicht ausgeführt, 1/WAHR: Makro wird ausgeführt).

Farbe: Bestimmt die Hintergrundfarbe des Makros (0: keine Hintergrundfarbe, 1: Hintergrundfarbe 0, 2: Hintergrundfarbe 1, ...)

Rahmen: Gibt die Rahmenart an (0: kein Rahmen, 1: einfacher Rahmen, 2: normaler 3D-Rahmen, 3: "inverser" 3D-Rahmen)

Beliebige Parameter können ausgelassen werden, dann wird die jeweilige Einstellung nicht verändert.

Wird kein Parameter bzw. nur der Name angegeben, so erscheint das Auswahlfenster für die Parametereinstellung (wie bei Mausklick) für das angegeben bzw. aktive Objekt.

1.175 OBJEKTPOS(Name;X;Y;Breite;Höhe)

OBJEKTPOS(Name;X;Y;Breite;Höhe)

(Englisch: OBJECTPOS)

Ändert die Position und/oder Größe des aktiven Objekts bzw. des Objekts Name.

Name: Ist der Name des Objekts. Wird er ausgelassen, so wird das aktuell ausgewählte Objekt verwandt. Ist kein Objekt ausgewählt, so erscheint eine Fehlermeldung.

X,Y,Breite,Höhe: Der jeweilige Parameter, auslassen oder -1 ändern die Einstellung nicht. Bei diesen Angaben handelt es sich nicht um Zellangaben (Objekte müssen ja nicht genau bei Zellgrenzen beginnen), sondern um ein kleineres Maß.

1.176 RÜCKGÄNGIG()

RÜCKGÄNGIG()

(Englisch: UNDO)

Dies entspricht <Bearbeiten-Rückgängig>.

1.177 SPEZIALOPTIONEN(Flags)

SPEZIALOPTIONEN(Flags)

(Englisch: SPECIALFLAGS)

Dieses Makro dient zum Einstellen einiger spezieller Optionen für die aktuelle Tabelle, z.Z ist nur eine vorhanden:

Flags:

0 normal öffnen

1 Beim Öffnen der Tabelle wird kein Fenster angezeigt, die Tabelle wird also nur in den Hintergrund geladen.

Hinweis: Die anderen Flags sind für spätere Erweiterungen gedacht und sollten stets auf 0 gesetzt werden (d.h. nur 0 oder 1 benutzen!)

Diese Einstellung wird bei den vorhandenen AutoÖffnen-Tabellen benutzt, die ja nur einen Menüpunkt hinzufügen und somit kein Fenster benötigen. Entsprechend können Sie eigene Makros 'in den Hintergrund' laden.

1.178 STOPAUFNAHME()

STOPAUFNAHME()

(Englisch: STOPRECORD)

Entspricht <Makro-Aufnahme stoppen> - Dieser Befehl sollte nicht in einem Makro verwendet werden, sondern dient nur zur Menüerstellung.

1.179 SYSINFO()

SYSINFO()

Entspricht <Hilfe-Info> und zeigt aktuelle System-Informationen an.

1.180 TCMAKRO(TCLib;Offset;Num1;Num2;Text)

TCMAKRO(TCLib;Offset;Num1;Num2;Text)

(Englisch: TCMACRO)

Hiermit können externe Makrobefehle eingebunden und aufgerufen werden. Dies ist eine einfache Möglichkeit, TurboCalc um weitere Funktionen zu erweitern. Dazu werden TurboCalc-Libraries benutzt. Einzelheiten dazu finden im Kapitel "Dateien".

TCLib: Dies ist der Name der TurboCalc-Library (als Text)

Offset: Gibt die Befehlsnummer an (somit können in einer Bibliothek mehrere Befehle vorhanden sein).

Num1, Num2, Text: Sind die möglichen Parameter für den Befehl (Num1 und Num sind Zahlenwerte, Text ist ein Text)

Die genaue Funktionsweise sowie die Parameterbelegung finden Sie in der Anleitung zu der jeweiligen TC-Bibliothek. Möchten Sie selbst eine eigene Bibliothek erstellen, so finden Sie Einzelheiten dazu im oben genannten Abschnitt über TCLibs.

1.181 ÜBER()

ÜBER()

(Englisch: ABOUT)

Dies entspricht <Datei-Über> und zeigt ein Fenster an, in dem Information über den Autor und das Programm zu finden sind.

1.182 WIEDERHERSTELLEN()

WIEDERHERSTELLEN()

(Englisch: REDO)

Dies entspricht <Bearbeiten-Wiederherstellen>.

1.183 Makro-Steuerung

Makro-Steuerung

Nun noch einige "Nur Makrofunktionen". Sie dienen der Steuerung während der Makroausführung und erlauben (bedingte) Verzweigung und Unterrouinen. Diese können nicht als ARexx-Befehl benutzt werden!

BEIFEHLER(ZELLE)
BIS(Bedingung)
FÜR(Variable;Anfang;Ende;Schritt)
FÜRBLOCK(Variable;Block;Flags)
MAKRO(...)
MSONST()
MWENN(Bedingung)
MWENNENDE()
NÄCHSTES()
ROUTINE(Zelle)
RÜCKSPRUNG([Zelle])
SCHLEIFE()
SCHRITT([Flag])
SOLANGE(Bedingung)
SPRINGE(Zelle)
SPRINGEWENN(Bedingung;Zelle)

1.184 BEIFEHLER(ZELLE)

BEIFEHLER(ZELLE)

(Englisch: ONERROR)

Damit kann eine Zelle bestimmt werden, die angesprungen wird, wenn ein Fehler auftritt. Die Fehlernummer kann mittels der Funktion LETZTERFEHLER (LASTERROR) ermittelt werden. Die Fehlerbehandlung muß mit einem RETURN() bzw. RETURN(Zelle) (RÜCKSPRUNG) beendet werden. Tritt vor einem solchen RETURN ein weiterer Fehler auf, so wird die Makroausführung beendet.

Zelle: Bestimmt die Zelle, zu der gesprungen wird, falls ein Fehler auftritt. Wird dieser Parameter ausgelassen, so wird die Fehleroutine ausgeschaltet.

Beispiel:

BEIFEHLER(X1) tritt ein Fehler auf, so wird zum Stelle X1 verzweigt

ONERROR() schaltet die Fehlerabfrage wieder aus.

Die (sehr einfache) Fehleroutine könnte wie folgt aussehen:

X1 =MELDUNG("Es trat der Fehler Nr."+TEXT(LETZTERFEHLER)+" auf!")

X2 =RETURN()

(siehe auch Beispieldatei)

1.185 BIS(Bedingung)

BIS(Bedingung)

(Englisch: UNTIL)

Dies ist eine Endbedingung zu SCHLEIFE (bzw. LOOP).

Ist die Bedingung falsch, so wird das Programm ab dem Befehl nach LOOP fortgesetzt, ansonsten wird der SOLANGE-Befehl einfach übersprungen und der nächste ausgeführt.

Einzelheiten siehe bei SCHLEIFE

Dies ist das Gegenstück zu SOLANGE

1.186 FÜR(Variable;Anfang;Ende;Schritt)

FÜR(Variable;Anfang;Ende;Schritt)

(Englisch: FOR)

Dies realisiert eine in fast allen Programmiersprachen vorhanden FOR...NEXT-Schleife. Dabei werden die folgenden Zeilen bis NÄCHSTES (oder NEXT) mehrmals durchlaufen.

Variable: Dies ist ein Text(!), der den Variablennamen angibt, der auf den jeweiligen Zählwert gesetzt wird. Damit kann innerhalb der Schleife der aktuelle Schleifenwert abgefragt werden. Bezüglich Variablen siehe NAMENFESTLEGEN. Dieser Parameter kann auch ausgelassen werden bzw. mit "" angegeben werden, dann wird keine Variable festgelegt.

Anfang: Anfangswert der Schleife

Endwert: Endwert der Schleife

Schritt: Schrittweite der Schleife. Dieser Wert wird jeweils zum aktuellen Wert (zu Beginn: Anfangswert) hinzuaddiert. Kann auch negativ sein. Wird er ausgelassen, so wird 1 angenommen.

Bei jedem NÄCHSTES wird der aktuelle Wert (wird zu Beginn auf den Anfangswert gesetzt) um Schritt erhöht und dann wie folgt verfahren:

Schritt>=0: Ist aktueller Wert<=Endwert, so wird beim Befehl unmittelbar nach FÜR fortgesetzt.

Schritt<0: Ist aktueller Wert>Endwert, so wird beim Befehl unmittelbar nach FÜR fortgesetzt.

Ansonsten wird die Schleife beendet und der Befehl nach NÄCHSTES ausgeführt.

Anmerkung: Statt NÄCHSTES kann auch SOLANGE(Bed) bzw. BIS(Bed) benutzt werden. Dadurch kann die Schleife noch mit einer weiteren Bedingung versehen werden. Beschreibung siehe bei SCHLEIFE.

Hinweis: Die FÜR-Schleife wird (im Gegensatz zu anderen Programmiersprachen) mindestens einmal ausgeführt - ein Test findet erst bei NÄCHSTES (bzw. BIS/SOLANGE) statt.

Beispiele:

=FÜR("abc";1;5)

=MELDUNG(text(abc))

=NÄCHSTES()

gibt nacheinander die Zahlen 1, 2, ... 5 aus

A1 =FÜR("abc";5;1;-2)

A2 =ANFRAGE(text(abc))

A3 =SOLANGE(A2)

A4 ...

gibt nacheinander 5, 3, 1 aus. Wird bei Anfrage auf Abbruch gedrückt, so wird die Schleife unterbrochen. In Zelle A4 (nach Ende der Schleife) ist abc auf "-1" gesetzt, falls kein vorzeitiger Abbruch durch SOLANGE stattfand!

=FÜR("");1;3)

=MELDUNG("Hallo")

=NÄCHSTES()

gibt 3 mal "Hallo" aus - eine Zählvariable wird hier nicht benötigt und somit auch nicht angegeben.

=FÜR("");0;0;0)

...

=BIS(...)

dies entspricht SCHLEIFE... BIS (wenn auch etwas umständlich...)

1.187 FÜRBLOCK(Variable;Block;Flags)

FÜRBLOCK(Variable;Block;Flags)

(Englisch: FORRANGE)

Eine Schleifenkonstruktion ähnlich FÜR, jedoch wird die Schleife statt für einen bestimmten Zahlenbereich für jede Zelle des Blockes ausgeführt:

Variable: Dies ist ein Text(!), der den Variablennamen angibt, der auf den jeweiligen Zelle gesetzt wird. Damit kann innerhalb der Schleife die aktuelle Zelle des Blockes abgefragt werden. Bezüglich Variablen siehe NAMENFESTLEGEN. Dieser Parameter kann auch ausgelassen werden bzw. mit "" angegeben werden, dann wird keine Variable festgelegt.

Block: gibt den Block an, der nacheinander "abgearbeitet" werden soll. Wird dieser Parameter ausgelassen, so wird der aktuell definierte Block der aktuellen Tabelle genommen.

Flags: Summe aus folgenden Zahlen:

1: Der Block wird Spaltenweise abgearbeitet (sonst: zeilenweise)

2: Der Block wird von rechts unten nach links oben abgearbeitet. (sonst: links oben ... rechts unten)

4: Der Zellcursor wird zu Beginn der Schleife auf die jeweilige Zelle gesetzt, ansonsten wird der Zellcursor nicht verändert. In beiden Fällen kann die aktuelle Zelle aus Variable ausgelesen werden, falls angegeben.

Beispiele:

=FÜRBLOCK("position";A1:C5)

=SCHREIBE(position+1;position)

=NÄCHSTES()

Erhöht alle Zellen des Bereichs A1:C5 um 1.

=FÜRBLOCK("");A1:C5;4)

=FARBEN(5)

=NÄCHSTES()

Färbt die Zellen des Bereichs A1:C5 rot (bzw. mit Farbe 5) ein. Der Parameter 4 gibt an, dass der Zellcursor entsprechend gesetzt werden soll. Deshalb wird bei FARBEN kein Zell-Parameter benötigt und der Variablenname kann so entfallen.

=FÜRBLOCK("position";A1:C5)

=FARBEN(5;;position)

=NÄCHSTES()

Hat die gleiche Funktion wie das Makro zuvor, da der Zellcursor jedoch nicht für jeden Schleifendurchlauf neu gesetzt werden muß, ist diese Funktion schneller.

1.188 MAKRO(...)

MAKRO(...)

(Englisch: MACRO)

Dieser Befehl macht nichts! Er erlaubt jedoch die Angabe eines Makronamens. Es empfiehlt sich, diesen Befehl als ersten Befehl jedes Makros zu schreiben und in Klammern den Titel des Makros anzugeben. So hat man dann immer den Überblick, um welches Makro es sich handelt.

Der Ausdruck in den Klammern wird ignoriert - es muß sich also um keine gültige Formel bzw. einen Text in Anführungszeichen handeln.

Hinweis: Diesen Befehl nicht mit dem Befehl zum Starten eines Makros verwechseln. Dafür bitte MAKROABSPIELEN (MACROPLAY) benutzen.

Beispiel:

MAKRO(TestMakro)

1.189 MSONST()

MSONST()

(Englisch: MELSE)

Dies leitet den Sonst-Zweig bei der MWENN-Abfrage ein, siehe dort.

1.190 MWENN(Bedingung)

MWENN(Bedingung)

(Englisch: MIF)

Mit dieser Kontrollstruktur können Anweisungen ähnlich SPRINGEWENN abhängig von Bedingungen ausgeführt werden:

=MWENN(Bedingung) =MIF(Condition)

...

=MSONST() =MELSE()

...

=MWENNENDE() =MENDIF()

Vorteil dieser Konstruktion ist, daß kein Sprung angegeben werden muß, sondern entweder (falls die Bedingung wahr ist) der Teil nach MWENN (bis MSONST bzw. MENDIF) ausgeführt wird oder übersprungen wird (falls Bedingung falsch). Im zweiten Fall wird der Teil zwischen MSONST und MWENNENDE (falls vorhanden) ausgeführt.

MIF...[MELSE]...MENDIF können natürlich auch (mehrfach) geschachtelt werden.

Hinweis: Jedes MIF muß mit MENDIF abgeschlossen werden! Die Angabe von MELSE (zwischen MIF und MENDIF) ist optional.

Beispiel:

=MWENN(wert(A1)=5)

= MELDUNG("Zelle A1 enthält 5")

=MWENNENDE()

Gibt die Meldung "A1 enthält 5" aus, falls dies zutrifft. Ansonsten wird diese Anweisung übersprungen.

=MIF(wert(A1)=5)

= MELDUNG("Zelle A1 enthält 5")

=MELSE

= MELDUNG("zelle A1 enthält nicht 5!")

=MENDIF

Wie zuvor, ansonsten wird jedoch der Text "A1 enthält nicht 5" ausgegeben.

1.191 MWENNENDE()

MWENNENDE()

(Englisch: MENDIF)

Dies beendet eine MWENN-Abfrage ein, siehe dort.

1.192 NÄCHSTES()

NÄCHSTES()

(English: NEXT)

Beenden FÜR und FÜRBLOCK-Schleifen, siehe dort.

1.193 ROUTINE(Zelle)

ROUTINE(Zelle)

(Englisch: CALL)

Verzweigt die Makroausführung an die Position Zelle, merkt sich jedoch die aktuelle Stelle, um dort nach einem RÜCKSPRUNG bzw. RETURN die Ausführung fortzusetzen. Der nächste Befehl wird also aus der Zelle Zelle ausgelesen und das Makro dort fortgesetzt.

Dieser Befehl ist also ähnlich SPRINGE, eignet sich jedoch zum Ausführen von Unterrouتين. Tritt in der Unterroutine ein RÜCKSPRUNG auf, so wird die Ausführung nach dem ROUTINE-Befehl fortgesetzt.

Zelle: Ist ein Zellbezug (etwa C10) oder ein Zellname (etwa Schleife), der angibt, an welcher Stelle das Makro fortgesetzt werden soll.

Hinweis: Normalerweise beziehen sich alle Zellbezüge automatisch auf die aktuelle Tabelle (und nicht auf die Makro-Tabelle). Da bei diesem Befehl jedoch meist innerhalb der Makro-Tabelle gesprungen werden soll, bezieht sich diese Zellangabe auf die Makrotabelle, unabhängig davon, welche Tabelle gerade aktiv ist. Auch die Funktion ZELLE gibt die Position relativ zu der Zelle an, in der sich der Makro-Befehl befindet (und nicht relativ zur aktuellen Cursorposition!)

Soll wirklich in eine andere Tabelle gesprungen werden, so kann dies durch Angabe eines AT-Bezugs (etwa @Makro;C5) realisiert werden.

1.194 RÜCKSPRUNG([Zelle])

RÜCKSPRUNG([Zelle])

(English: RETURN)

Dies sollte der letzte Befehl in jedem Makro sein - er beendet das Makro.

Befindet man sich (durch ROUTINE) in einer Unterroutine, so beendet er die entsprechende Unterroutine und kehrt zur Stelle zurück, an der die Unterroutine aufgerufen wurde.

Zelle: Wird Zelle angegeben, so wird statt zur Zeile nach dem Unterrouتينaufruf zu Zelle verzweigt (und weiterhin die Unterroutine beendet). Normalerweise wird dieser Parameter wohl ausgelassen.

1.195 SCHLEIFE()

SCHLEIFE()

(English: LOOP)

Dies realisiert zusammen mit BIS bzw. SOLANGE (UNTIL bzw. WHILE) eine Schleife, d.h. ein Teil, der mehrfach durchlaufen wird: Der Teil zwischen SCHLEIFE und BIS bzw. SOLANGE wird dabei mindestens einmal durchlaufen. Trifft Sie dann auf SOLANGE und ist diese Bedingung WAHR (bzw. bei BIS FALSCH), so wird der Teil ab SCHLEIFE wiederholt und zwar bis die Bedingung FALSCH (bzw. bei BIS WAHR) ist.

Da die Schleifenkonstruktion ohne Zellverweise (wie etwa bei SPRINGEWENN) auskommt, erhöht es die Lesbarkeit und die Fehlersicherheit. Auch ist es so besser verschiebbar. Wenn immer möglich, sollten Sie also anstelle von SPRINGEWENN die SCHLEIFE benutzen.

Hinweis: Springen Sie innerhalb einer Schleife mittels RETURN() zurück, so erscheint eine Fehlermeldung!

1. Beispiel:

```
=PUT(1;A1)
=SCHLEIFE()
= PUT(A1+1;A1)
=SOLANGE(A1<10)
```

2. Beispiel:

```
=PUT(1;A1)
=SCHLEIFE()
= PUT(A1+1;A1)
=BIS(A1=11)
```

Beide Beispiele schreiben in die Zelle A1 den Wert 1 und erhöhen ihn dann jeweils um 1, und zwar:

1. solange er kleiner als 10 ist
2. bis er genau 11 ist.

(In diesem Fall führt dies beides auf den gleichen Wert).

Tip: Wie sie vielleicht gesehen haben, kann man zwischen dem Gleichheitszeichen und dem Befehl Leerzeichen lassen. Dies erhöht (etwa bei geschachtelten Schleifen) die Lesbarkeit und ist zu empfehlen.

Tip: Es ist immer BIS und SOLANGE als Schleifenbedingung möglich, meist ist jedoch eine Bedingung einfacher und übersichtlicher.

Es sind natürlich auch mehrfach geschachtelte Schleifen möglich, hierzu ein einfaches Beispiel:

```
=PUT(1;A1)
=PUT(1;A3)
=LOOP()
= PUT(1;A2)
= LOOP()
= PUT(A2+1;A2)
= PUT(A3+1;A3)
= UNTIL(A2=5)
= PUT(A1+1;A1)
=UNTIL(A1=5)
```

Dieses Beispielprogramm zählt die A1 von 1 auf 5 und (innerhalb dieses Hochzählens) jeweils A2 von 1 auf 5. Bei jedem Schritt wird A3 um eins erhöht.

1.196 SCHRITT([Flag])

SCHRITT([Flag])

(Englisch: STEP)

Dies schaltet den Einzelschrittmodus ein (falls Flag fehlt bzw. ungleich Null ist) bzw. aus (bei 0 bzw. FALSCH)

Dieser Modus macht vor jedem Befehl eine Pause und zeigt den jeweiligen Befehl an. Dies eignet sich somit gut zur Fehlersuche in Makros, siehe auch entsprechendes Kapitel bei "Makro-Befehle".

1.197 SOLANGE(Bedingung)

SOLANGE(Bedingung)

(Englisch: WHILE)

Dies ist eine Endbedingung zu SCHLEIFE (bzw. LOOP).

Ist die Bedingung wahr, so wird das Programm ab dem Befehl nach LOOP fortgesetzt, ansonsten wird der SOLANGE-Befehl einfach übersprungen und der nächste ausgeführt.

Einzelheiten siehe bei SCHLEIFE

Dies ist das Gegenstück zu BIS.

1.198 SPRINGE(Zelle)

SPRINGE(Zelle)

(Englisch: GOTO)

Verzweigt die Makroausführung an die Position Zelle. Der nächste Befehl wird also aus der Zelle Zelle ausgelesen und das Makro dort fortgesetzt.

Zelle: Ist ein Zellbezug (etwa C10) oder ein Zellname (etwa Schleife), der angibt, an welcher Stelle das Makro fortgesetzt werden soll.

Hinweis: Normalerweise beziehen sich alle Zellbezüge automatisch auf die aktuelle Tabelle (und nicht auf die Makro-Tabelle). Da bei diesem Befehl jedoch meist innerhalb der Makro-Tabelle gesprungen werden soll, bezieht sich diese Zellangabe auf die Makrotabelle, unabhängig davon, welche Tabelle gerade aktiv ist. Auch die Funktion ZELLE gibt die Position relativ zu der Zelle an, in der sich der Makro-Befehl befindet (und nicht relativ zur aktuellen Cursorposition!)

Soll wirklich in eine andere Tabelle gesprungen werden, so kann dies durch Angabe eines AT-Bezugs (etwa @Makro;C5) realisiert werden.

1.199 SPRINGEWENN(Bedingung;Zelle)

SPRINGEWENN(Bedingung;Zelle)

(Englisch: IFGOTO)

Dies ist eine bedingte Verzweigung. Ist die Bedingung WAHR (bzw. ungleich Null), so wird die Ausführung des Makros an der Stelle Zelle fortgesetzt, sonst in der nächsten Zeile.

Bedingung: Beliebiger Wahrheitswert (mit =, <>, <, <=, >, >=... siehe auch "Formeln - Wahrheitswerte").

Zelle: Gibt die Zelle an, an der die Ausführung fortgesetzt werden soll, wenn Bedingung WAHR ist. (Einzelheiten siehe bei SPRINGE!)

Dies entspricht also SPRINGE, falls Bedingung WAHR ist (siehe auch dort).

Beispiel:

SPRINGEWENN(WAHR;C5)

entspricht SPRINGE(C5), (da WAHR immer richtig ist)

folgender Programmausschnitt führt die Befehle in den Zeilen A11 bis A13 zehnmal aus. (Es wird 10 in die Zelle B1 geschrieben und dann immer um eins erniedrigt (9,8, ...). Solange B1 größer als 0 ist, wird dies wiederholt.)

So kann also relativ einfach eine Schleife in TurboCalc programmiert werden.

Zelle Makro-Befehl

A10 =PUT(10;B1)

A11

A12 =PUT(B1;CELLABS(B1;3))

A13

A14 =PUT(B1-1;B1)

A15 =SPRINGEWENN(B1>0;A11)

Der Befehl in Zelle A12 bewirkt, daß die Zelle mit der Spalte 3 und der Zeile B1 (also 1 bis 10!) auf 1 bis 10 gesetzt wird: die Zellen C1 bis C10 bekommen also den Inhalt 1 bis 10!

1.200 spezielle ARExx-Befehle

spezielle ARExx-Befehle

Nun noch einige spezielle ARExx-Befehle (neben den Standard ARExx/Makro-Befehlen), die nur als ARExx-Befehl sinnvoll sind und deswegen nicht als Makro-Befehl aufgerufen werden sollten. (Da die meisten Befehle zum Auslesen von Werten und anderen Angaben aus Tabellen benötigt werden und dies aus einem Makro direkt mittels Zellangabe, etc gemacht werden kann, ist dies auch weder nötig noch sinnvoll.)

KOMMENTAR

LIESWERT [Wert]

LIESFORMEL [Zelle]

LIESCURSORPOS

1.201 KOMMENTAR

KOMMENTAR

(Englisch: REM oder auch ** bzw. --)

Dieser Befehl macht nichts, der ggf. dahinterstehende Text wird als Kommentar aufgefaßt. Dies ist nur für Makro-Skript-Dateien sinnvoll (siehe beim Befehl START), bei Makros kann der Kommentar als normaler Text geschrieben werden.

1.202 LIESWERT [Wert]

LIESWERT [Wert]

(Englisch: GETVALUE)

Dieser Befehl wird dazu benutzt Werte (Zahlen, Texte...) von TurboCalc auszulesen und an das ARExx-Skript zu übergeben. Dort kann er dann unter RESULT angesprochen werden.

Wert: Kann eine beliebige Formel (also einfacher Wert, Zellverweis, Funktionen...) sein. Nur ein reiner Text ist nicht möglich. Fehlt diese Angabe, so wird der Inhalt der aktuellen Zelle ausgelesen und angezeigt.

Hinweis: Normalerweise wird dieser Befehl wohl hauptsächlich dazu benutzt, den Inhalt von Zellen auszulesen (Rechnen kann ja normalerweise direkt von ARExx aus ausgeführt werden:). Siehe dazu die ersten beiden Beispiele.

Tip: Achten Sie darauf, daß zu Beginn Ihres ARExx-Skripts der Befehl "Options Results" steht. Dies ist ein interner ARExx-Befehl, der die Rückgabe von Werten erlaubt.

Beispiel:

LIESWERT A1

SAY RESULT

liest den Wert der Zelle A1 und gibt ihn aus.

"AUSWÄHLEN A1"

LIESWERT

SAY RESULT

entsprechend, hier liest LIESWERT ohne Parameter die aktuelle Zelle aus.

GETVALUE "5+5"

ergibt den Wert 10 in RESULT (nicht so sinnvoll, aber:)

GETVALUE "SUMME(A1:A10)"

berechnet die Summe der Zellen A1 bis A10

1.203 LIESFORMEL [Zelle]

LIESFORMEL [Zelle]

(Englisch: GETFORMULA)

Dieser Befehl wird dazu benutzt, Formeln aus einer Zelle von TurboCalc auszulesen und an das ARexx-Skript zu übergeben. Dort kann er dann unter RESULT angesprochen werden.

Zelle: Kann ein beliebiger Zellbezug sein. Fehlt diese Angabe, so wird die Formel der aktuellen Zelle ausgelesen und angezeigt.

Enthält die aktuelle Zelle keine Formel, so wird "" (d.h. ein Leertext) zurückgegeben.

Tip: Achten Sie darauf, daß zu Beginn Ihres ARexx-Skripts der Befehl "Options Results" steht. Dies ist ein interner ARexx-Befehl, der die Rückgabe von Werten erlaubt.

Beispiel:

PUT "=A1+A2" A3

LIESFORMEL A3

SAY RESULT

liest die Formel aus der Zelle A3 (die zuvor als =A1+A2. definiert wurde): =A1+A2

"AUSWÄHLEN A3"

GETFORMULA

SAY RESULT

entsprechend.

1.204 LIESCURSORPOS

LIESCURSORPOS

(Englisch: GETCURSORPOS)

Dieser Befehl liest die aktuelle Zelle bzw. den aktuellen Block der ausgewählten Tabelle aus und speichert es unter RESULT als normaler Bezug (A1 oder C5:H7) ab.

Hinweis: Achten Sie darauf, daß zu Beginn Ihres ARexx-Skripts der Befehl "Options Results" steht. Dies ist ein interner ARexx-Befehl, der die Rückgabe von Werten erlaubt.

Tip: Dies kann dazu benutzt werden, die Cursorposition zwischenzuspeichern.

Beispiel:

```
/* alte Position auslesen */  
GETCURSORPOS  
oldpos = RESULT  
/* Hier kommen einige Befehle */  
/* .... */  
/* AltePosition setzen */  
"SELECT" oldpos
```

1.205 Komplettes Inhaltsverzeichnis

Allgemeines

Zahlen

Wahrheitswerte

Texte

Zelle/Bereich

Auslassen von Parametern

Auslassen aller Parameter

Rückgabewerte von Befehlen

Menü-Befehle

Blockbefehle

AUSSCHNEIDEN([Block])

BEZUGTYP(Flag[;Block])

BEZUGVERSCHIEBEN(X;Y;Flag;Block)

DATENEINFÜGEN(Modus,[Block])

EINFÜGEN([Block])

ENTFERNEN(Data,[Block])

FÜLLEN(Mode,[Block])

HINZUFÜGEN(Data,[Block])

KOPIEREN([Block])

LÖSCHEN(Data,[Block])

MMAL(Block1;Block2;ErgBlock)

MMINUS(Block1;Block2;ErgBlock)

MPLUS(Block1;Block2;ErgBlock)

REIHEBERECHNEN(Typ;Inkrement;Spalten;Bereich)

SORTIEREN(Aufsteigend;Richtung;Zelle;Bereich)

SPRACHE(Modus;Block)

TRANSPONIEREN([Block])

Formatierungen

ANZEIGEN(Zeile;[Block])
AUSRICHTUNG([Hor];[Vert];[Block])
FARBEN([Farbe1];[Farbe2];[Block])
FIXIEREN(Zelle)
MUSTER(Zahl;[Block])
KASTEN(Links;Rechts;Oben;Unten;[Block])
RAHMEN(Links;Rechts;Oben;Unten;[Block])
SCHRIFT([Num];[Zeichensatz];[Block])
SPALTENBREITE(Breite;[Block])
STDSCHRIFT(Zeichensatz)
STILWECHSELN(Num;[Block])
VERBERGEN(Zeile;[Block])
ZAHLENFORMAT(Format;[Block])
ZEILENHÖHE(Höhe;[Block])
ZELLSCHUTZ([Schreiben];[Formel];[Block])
Cursorsteuerung
AKTIVEZELLE()
AUSWÄHLEN([Block])
INSPALTE(Spalte)
INZEILE(Zeile)
LINKS(Num
SPALTE(Spalte)
LETZTESPALTE()
SUCHEN(Text;Teil;Groß;Spalten;Bereich)
ZEILE(Zeile)
LETZTEZEILE()
Eingabe
AMIGAGUIDE(Datei;Kommando)
ANFRAGE(Text[;Titel])
DIALOG(Dialog;Hook)
DLGWEITER(Flag)
EINGABE(Text[;Titel];[Zelle])
HILFE(Num;Datei)
MELDUNG(Text[;Titel])
PAUSE(Zeit)
REQPARA(X;Y;Oktext;Abbruchtext;Zeit)
SCHREIBE(Inhalt[;Zelle])
SIGNAL()
Laden/Speichern

ABLAGENLESEN([Block])
ABLAGESCHREIBEN([Block])
BLOCKSPEICHERN([Block];[Name])
CSVBLOCKSPEICHERN([Block];[Name];[Trennzeichen])
CSVEINFÜGEN([Block];[Name];[Trennzeichen])
CSVLADEN([Name];[Trennzeichen])
CSVSPEICHERN([Name];[Trennzeichen])
KONFIGLADEN()
KONFIGSPEICHERN()
LADEN([Name])
PROCALCEINFÜGEN([Block];[Name])
PROCALCLADEN([Name])
SPEICHERN([Name])
SPEICHERNALS([Name])
SYLKBLOCKSPEICHERN([Block];[Name])
SYLKEINFÜGEN([Block];[Name])
SYLKLADEN([Name])
SYLKSPEICHERN([Name])
TCDEINFÜGEN([Block];[Name])

Datenbank

DATENBANK([Bereich])
DBEXTRAHIEREN([Zelle])
DBLETZTER(Zelle)
DBLÖSCHEN()
DBSORTIEREN(Aufsteigend;[Zelle])
DBSUCHEN([Zelle])
KRITERIEN([Bereich])

Optionen

BILDAUFBAU(Modus)
DARSTELLUNG(Titel;Raster;Werkzeugl.;Formeln;Null;Cursorm,,)
DEFPUBSCREEN(Bildschirm)
DRUCKBEREICH(Aktivieren;[Bereich])
DRUCKFORMAT(LR;RR;OR;UR;Stil;KZeile;KText;FZeile;FText;Titel;Raster)
GLOBALEOPTIONEN()
LADEOPTIONEN(KeinAutoMakro;Verbergen;Passwort)
LÄNDEROPTIONEN(ZF1;ZF2;DF;Währung;WPräfix;WSuffix;Inch)
PAPIERFORMAT()
SCHUTZOPTIONEN()
SEITENVORSCHUB(Flag)

SHANGHAI(Modus)
SMARTREFRESH(Flag)
TABELLENOPTIONEN(MaxBreite;MaxHöhe;Berechnung;Return;Richtung;Icons)
TASTE(Taste;Befehl)
ZWISCHENABLAGE(Einheit;Trennzeichen;Anführungszeichen)
ZWISCHENSPEICHERN(Flag)
Menübefehle
LÖSCHEMENÜPUNKT(Titel;Punkt)
LÖSCHEMENÜTITEL(Titel)
LÖSCHEMENÜUNTER(Titel;Punkt;Unter)
NEUERMENÜPUNKT(Name;Befehl;[Titel;Punkt])
NEUERMENÜTITEL(Name;[Titel])
NEUERMENÜUNTER(Name;Befehl;[Titel;Punkt;Unter])
NEUESMENÜ()
ZEIGEMENÜ()
Steuerung
AUSFÜHREN(Datei;Parameter;[Fenster])
BEENDEN([Flag])
BLOCKNAMENFESTLEGEN(Name;Block)
DRUCKEN(Drucker;File;NLQ;Bereich;Seite1;Seite2;LPI;Farbig;Pause;Größe;Breite;Höhe)
IFFDRUCKEN(Datei;Breite;Höhe;Tiefe;Block)
MAKROABSPIELEN(Zelle)
NAMENFESTLEGEN(Name;Wert)
NAMENLÖSCHEN(Name)
NEUBERECHNEN([Modus])
NEUETABELLE(Name)
RUN(Datei;Parameter;[Fenster])
RX(Datei;Port)
START(Dateiname)
TABELLEEINBLENDEN(Tabellenname;Fensternummer)
TABELLESCHLIEßEN(Sofort)
TABELLEVERBERGEN(Tabellenname;Fensternummer)
UNGEÄNDERT()
WÄHLETABELLE(Name[;FensterNummer])
Bildschirm
ALTEFARBEN()
BILDSCHIRM(Breite;Höhe;Tiefe;Modus)
FARBEÄNDERN(Farbe;Rot;Grün;Blau)
FENSTERAKTIVIEREN()

FENSTERANORDNEN()
FENSTERANORDNEN2()
FENSTERGRÖßE(Breite;Höhe)
FENSTERNACHHINTEN()
FENSTERNACHVORN()
FENSTERPOSITIONIEREN(X;Y;Breite;Höhe)
FENSTERVERSCHIEBEN(X;Y)
IKONIFIZIEREN()
STDFARBEN()
ZEICHENSATZ(Zeichensatz;Modus)
Sonstige Befehle
AUFNAHME()
COMMAND()
DIAEINBLENDEN()
DRUCKVORSCHAU()
EDIT()
FORMELEINFÜGEN()
MAKROEINFÜGEN()
NAMENEINFÜGEN()
NEUESFENSTER()
OBJEKT(Name;Typ;Text;Wert1;Wert2;Block)
OBJEKTAUSWAHL(Name)
OBJEKTKLASSE(Name)
OBJEKTPARA(Name;Makro;Makroaktivierung;Farbe;Rahmen)
OBJEKTPOS(Name;X;Y;Breite;Höhe)
RÜCKGÄNGIG()
SPEZIALOPTIONEN(Flags)
STOPAUFNAHME()
SYSINFO()
TCMAKRO(TCLib;Offset;Num1;Num2;Text)
ÜBER()
WIEDERHERSTELLEN()
Makro-Steuerung
BEIFEHLER(ZELLE)
BIS(Bedingung)
FÜR(Variable;Anfang;Ende;Schritt)
FÜRBLOCK(Variable;Block;Flags)
MAKRO(...)
MSONST()

MWENN(Bedingung)
MWENNENDE()
NÄCHSTES()
ROUTINE(Zelle)
RÜCKSPRUNG([Zelle])
SCHLEIFE()
SCHRITT([Flag])
SOLANGE(Bedingung)
SPRINGE(Zelle)
SPRINGEWENN(Bedingung;Zelle)
spezielle ARexx-Befehle
KOMMENTAR
LIESWERT [Wert]
LIESFORMEL [Zelle]
LIESCURSORPOS

1.206 Index

A

ABLAGENLESEN([Block])
ABLAGESCHREIBEN([Block])
AKTIVEZELLE()
Allgemeines
ALTEFARBEN()
AMIGAGUIDE(Datei;Kommando)
ANFRAGE(Text[;Titel])
ANZEIGEN(Zeile;[Block])
AUFNAHME()
AUSFÜHREN(Datei;Parameter;[Fenster])
Auslassen aller Parameter
Auslassen von Parametern
AUSRICHTUNG([Hor];[Vert];[Block])
AUSSCHNEIDEN([Block])
AUSWÄHLEN([Block])

B

BEENDEN([Flag])
befindet).
BEIFEHLER(ZELLE)
BEZUGTYP(Flag;[Block])

BEZUGVERSCHIEBEN(X;Y;Flag;Block)

BILDAUFBAU(Modus)

Bildschirm

BILDSCHIRM(Breite;Höhe;Tiefe;Modus)

BIS(Bedingung)

Blockbefehle

BLOCKNAMENFESTLEGEN(Name;Block)

BLOCKSPEICHERN([Block];[Name])

C

COMMAND()

CSVBLOCKSPEICHERN([Block];[Name];[Trennzeichen])

CSVEINFÜGEN([Block];[Name];[Trennzeichen])

CSVLADEN([Name];[Trennzeichen])

CSVSPICHERN([Name];[Trennzeichen])

Cursorsteuerung

D

DARSTELLUNG(Titel;Raster;Werkzeugl.;Formeln;Null;Cursorm,..)

Datenbank

DATENBANK([Bereich])

DATENEINFÜGEN(Modus;[Block])

DBEXTRAHIEREN([Zelle])

DBLETZTER(Zelle)

DBLÖSCHEN()

DBSORTIEREN(Aufsteigend;[Zelle])

DBSUCHEN([Zelle])

DEFPUBSCREEN(Bildschirm)

DIAEINBLENDEN()

DIALOG(Dialog;Hook)

DLGWEITER(Flag)

DRUCKBEREICH(Aktivieren;[Bereich])

DRUCKEN(Drucker;File;NLQ;Bereich;Seite1;Seite2;LPI;Farbig;Pause;Größe;Breite;Höhe)

DRUCKFORMAT(LR;RR;OR;UR;Stil;KZeile;KText;FZeile;FText;Titel;Raster)

DRUCKVORSCHAU()

E

EDIT()

EINFÜGEN([Block])

Eingabe

EINGABE(Text[;Titel];[Zelle])

ENTFERNEN(Data;[Block])

F

FARBEN([Farbe1];[Farbe2];[Block])

FARBEÄNDERN(Farbe;Rot;Grün;Blau)

FENSTERAKTIVIEREN()

FENSTERANORDNEN()

FENSTERANORDNEN2()

FENSTERGRÖßE(Breite;Höhe)

FENSTERNACHHINTEN()

FENSTERNACHVORN()

FENSTERPOSITIONIEREN(X;Y;Breite;Höhe)

FENSTERVERSCHIEBEN(X;Y)

FIXIEREN(Zelle)

Formatierungen

FORMELEINFÜGEN()

FÜLLEN(Mode;[Block])

FÜR(Variable;Anfang;Ende;Schritt)

FÜRBLOCK(Variable;Block;Flags)

G

GLOBALEOPTIONEN()

H

HILFE(Num;Datei)

HINZUFÜGEN(Data;[Block])

HOCH(Num)

I

IFFDRUCKEN(Datei;Breite;Höhe;Tiefe;Block)

IKONIFIZIEREN()

INSPALTE(Spalte)

INZEILE(Zeile)

K

KASTEN(Links;Rechts;Oben;Unten;[Block])

KOMMENTAR

KONFIGLADEN()

KONFIGSPEICHERN()

KOPIEREN([Block])

KRITERIEN([Bereich])

L

LADEN([Name])

Laden/Speichern

LADEOPTIONEN(KeinAutoMakro;Verbergen;Passwort)

LETZTESPALTE()
LETZTEZEILE()
LIESCURSORPOS
LIESFORMEL [Zelle]
LIESWERT [Wert]
LINKS(Num)
LÄNDEROPTIONEN(ZF1;ZF2;DF;Währung;WPräfix;WSuffix;Inch)
LÖSCHEMENÜPUNKT(Titel;Punkt)
LÖSCHEMENÜTITEL(Titel)
LÖSCHEMENÜUNTER(Titel;Punkt;Unter)
LÖSCHEN(Data;[Block])

M

MAKRO(...)
Makro-Steuerung
MAKROABSPIELEN(Zelle)
MAKROEINFÜGEN()
MELDUNG(Text[;Titel])
Menü-Befehle
Menübefehle
MMAL(Block1;Block2;ErgBlock)
MMINUS(Block1;Block2;ErgBlock)
MPLUS(Block1;Block2;ErgBlock)
MSONST()
MUSTER(Zahl;[Block])
MWENN(Bedingung)
MWENNENDE()

N

NAMENEINFÜGEN()
NAMENFESTLEGEN(Name;Wert)
NAMENLÖSCHEN(Name)
NEUBERECHNEN([Modus])
NEUERMENÜPUNKT(Name;Befehl;[Titel;Punkt])
NEUERMENÜTITEL(Name;[Titel])
NEUERMENÜUNTER(Name;Befehl;[Titel;Punkt;Unter])
NEUESFENSTER()
NEUESMENÜ()
NEUETABELLE(Name)

NÄCHSTES()

O

OBJEKT(Name;Typ;Text;Wert1;Wert2;Block)

OBJEKTAUSWAHL(Name)

OBJEKTKLASSE(Name)

OBJEKTPARA(Name;Makro;Makroaktivierung;Farbe;Rahmen)

OBJEKTPOS(Name;X;Y;Breite;Höhe)

Optionen

P

PAPIERFORMAT()

PAUSE(Zeit)

PROCALCEINFÜGEN([Block];[Name])

PROCALCLADEN([Name])

R

RAHMEN(Links;Rechts;Oben;Unten;[Block])

RECHTS(Num)

REIHEBERECHNEN(Typ;Inkrement;Spalten;Bereich)

REQPARA(X;Y;Oktext;Abbruchtext;Zeit)

ROUTINE(Zelle)

RUN(Datei;Parameter;[Fenster])

RUNTER(Num)

RX(Datei;Port)

RÜCKGÄNGIG()

RÜCKSPRUNG([Zelle])

Rückgabewerte von Befehlen

S

SCHLEIFE()

SCHREIBE(Inhalt[;Zelle])

SCHRIFT([Num];[Zeichensatz];[Block])

SCHRITT([Flag])

SCHUTZOPTIONEN()

SEITENVORSCHUB(Flag)

SHANGHAI(Modus)

SIGNAL()

SMARTREFRESH(Flag)

SOLANGE(Bedingung)

Sonstige Befehle

SORTIEREN(Aufsteigend;Richtung;Zelle;Bereich)

SPALTE(Spalte)

SPALTENBREITE(Breite;[Block])

SPEICHERN([Name])

SPEICHERNALS([Name])

SPEZIALOPTIONEN(Flags)

spezielle ARexx-Befehle

SPRACHE(Modus;Block)

SPRINGE(Zelle)

SPRINGEWENN(Bedingung;Zelle)

START(Dateiname)

STDFARBEN()

STDSCHRIFT(Zeichensatz)

Steuerung

STILWECHSELN(Num;[Block])

STOPAUFNAHME()

SUCHEN(Text;Teil;Groß;Spalten;Bereich)

SYLKBLOCKSPEICHERN([Block];[Name])

SYLKEINFÜGEN([Block];[Name])

SYLKLADEN([Name])

SYLKSPEICHERN([Name])

SYSINFO()

T

TABELLEEINBLENDEN(Tabellenname;Fensternummer)

TABELLEOPTIONEN(MaxBreite;MaxHöhe;Berechnung;Return;Richtung;Icons)

TABELLESCHLIEßEN(Sofort)

TABELLEVERBERGEN(Tabellenname;Fensternummer)

TASTE(Taste;Befehl)

TCDEINFÜGEN([Block];[Name])

TCMAKRO(TCLib;Offset;Num1;Num2;Text)

Texte

TRANSPONIEREN([Block])

TurboCalc by Michael Friedrich

U

UNGEÄNDERT()

V

VERBERGEN(Zeile;[Block])

W

Wahrheitswerte

WIEDERHERSTELLEN()

WÄHLETABELLE(Name[;FensterNummer])

Z

Zahlen

ZAHLENFORMAT(Format;[Block])

ZEICHENSATZ(Zeichensatz;Modus)

ZEIGEMENÜ()

ZEILE(Zeile)

ZEILENHÖHE(Höhe;[Block])

Zelle/Bereich

ZELLSCHUTZ([Schreiben];[Formel];[Block])

ZWISCHENABLAGE(Einheit;Trennzeichen;Anführungszeichen)

ZWISCHENSPEICHERN(Flag)

Ü

ÜBER()
