

Funktionen

COLLABORATORS

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

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1 Funktionen	1
1.1 TurboCalc by Michael Friedrich	1
1.2 Datentypen	1
1.3 Operatoren	3
1.4 Mathematikfunktionen	3
1.5 ABRUNDEN (Zahl;Stellen)	5
1.6 ABS(Zahl)	5
1.7 ARCCOS(Zahl)	5
1.8 ARCSIN(Zahl)	6
1.9 ARCTAN(Zahl)	6
1.10 AUFRUNDEN(Zahl;Stellen)	6
1.11 BOGEN(Zahl)	7
1.12 COS(Zahl)	7
1.13 COSHYP(Zahl)	7
1.14 EXP(Zahl)	7
1.15 EUKLID1 (Zahl1;Zahl2)	8
1.16 EUKLID2 (Zahl1;Zahl2)	8
1.17 FAKULTÄT(Zahl)	8
1.18 GANZZAHL(Zahl)	8
1.19 GERADE(Zahl)	9
1.20 GGT(Zahl1;Zahl2)	9
1.21 GRENZE(X;a;b;W1;W2;W3)	9
1.22 HOCH(Basis;Exponent)	10
1.23 HOCH10(Exponent)	10
1.24 KEHRWERT(Zahl)	10
1.25 KGV(Zahl1;Zahl2)	10
1.26 KOMBINATIONEN(n;k)	11
1.27 KUM([Zahl1;]Zahl2)	11
1.28 KÜRZEN(Zahl)	11
1.29 LG(Zahl)	12

1.30 LN(Zahl)	12
1.31 LOG(Zahl)	12
1.32 LOG10(Zahl)	12
1.33 MINVERS(Zahl;Modulo)	12
1.34 NACHKOMMA(Zahl)	13
1.35 OBERGRENZE(Zahl;Schritt)	13
1.36 PI()	14
1.37 QUADRAT(Zahl)	14
1.38 REST(Zahl1;Zahl2)	14
1.39 RUNDEN(Zahl;Stellen)	14
1.40 SIN(Zahl)	15
1.41 SINHYP(Zahl)	15
1.42 TAN(Zahl)	15
1.43 TANHYP(Zahl)	16
1.44 UNGERADE(Zahl)	16
1.45 UNTERGRENZE(Zahl;Schritt)	16
1.46 VARIATIONEN(n;k)	17
1.47 VORZEICHEN(Zahl)	17
1.48 WINKEL(Zahl)	17
1.49 WURZEL(Zahl[;Basis])	18
1.50 ZUFALLSBEREICH(Min;Max)	18
1.51 ZUFALLSZAHL()	18
1.52 ZWEIFAKULTÄT(Zahl)	19
1.53 Wahrheitswert-Funktionen	19
1.54 ÄHNLICH(Text1;Text2)	20
1.55 ENTHÄLT(Text1;Text2)	20
1.56 FALSCH()	20
1.57 IDENTISCH(Text1;Text2)	20
1.58 ISTDATUM(Wert)	21
1.59 ISTFEHLER(Bezug)	21
1.60 ISTGERADE(Zahl)	21
1.61 ISTLEER(Bezug)	21
1.62 ISTTEXT(Wert)	22
1.63 ISTUNGERADE(Zahl)	22
1.64 ISTZAHL(Wert)	22
1.65 ISTZEIT(Wert)	22
1.66 MUSTERVERGLEICH(Text;Muster)	23
1.67 NICHT(Wert)	23
1.68 ODER(Wert1;Wert2;)	23

1.69 UND(Wert1;Wert2;)	24
1.70 VERGLEICH(Text1;Text2)	25
1.71 WAHR()	25
1.72 WENN(Bedingung; Wert1; Wert2)	25
1.73 XOR(Wert1;Wert2;)	26
1.74 Text-Funktionen	26
1.75 Plus: +	27
1.76 BEHALTEN(String;Liste)	28
1.77 BERECHNEN(Formel)	28
1.78 CODE(Text)	28
1.79 BIN(Zahltext)	28
1.80 CONV2INT(Zahltext;Basis)	29
1.81 CONV2TEXT(Zahl;Basis)	29
1.82 ERSETZEN(String;Text1[;Text2])	29
1.83 GLÄTTEN(Text)	30
1.84 GROSS(Text)	30
1.85 GROSS2(Text)	30
1.86 HEX(Zahltext)	30
1.87 INTEXT(String;Muster[;Pos])	31
1.88 KLEIN(Text)	31
1.89 KOMPRIMIEREN(String[;Liste])	31
1.90 LÄNGE(Text)	32
1.91 LINKS(Text;Anzahl)	32
1.92 MITTE(Text;Zahl1;Zahl2)	32
1.93 NACHBIN(Zahl)	33
1.94 NACHHEX(Zahl)	33
1.95 NACHOCT(Zahl)	33
1.96 OCT(Zahltext)	33
1.97 RECHTS(Text;Anzahl)	34
1.98 RÖMISCH(Zahl)	34
1.99 SÄUBERN(Text)	34
1.100SCHIEBENL(Text)	35
1.101SCHIEBENR(Text)	35
1.102SPIEGELN(Text)	35
1.103TEIL(Text;Zahl1;Zahl2)	35
1.104TEXT(Data[;Format])	35
1.105WERT(Text)	36
1.106WIEDERHOLEN(Text; Anzahl)	36
1.107WORT(Text;Num)	37

1.108 WÖRTER(Text)	37
1.109 ZEICHEN(asc-code)	37
1.110 Datum- und Zeit-Funktionen	38
1.111 Plus: +, Minus: -	38
1.112 DATUM(Jahr;Monat;Tag)	39
1.113 DATWERT(Text)	39
1.114 HEUTE()	40
1.115 JAHR(Datum)	40
1.116 JETZT()	40
1.117 MINUTE(Zeit)	41
1.118 MINUTEN(Zeit)	41
1.119 MONAT(Datum)	41
1.120 MONATE(Datum)	42
1.121 MONATTEXT(Datum[:Abkürzung])	42
1.122 SEKUNDE(Zeit)	42
1.123 SEKUNDEN(Zeit)	43
1.124 STUNDE(Zeit)	43
1.125 STUNDEN(Zeit)	43
1.126 TAG(Datum)	43
1.127 TAGTEXT(Datum[:Abkürzung])	44
1.128 WOCHENTAG(Datum)	44
1.129 ZEIT(Stunde;Minute;Sekunde)	44
1.130 ZEITWERT(Text)	45
1.131 Tabellenfunktionen	45
1.132 ANZAHL(Bereich[:])	46
1.133 ANZAHL2(Bereich[:])	46
1.134 ERSTER(Bereich;...)	47
1.135 LETZTER(Bereich;...)	47
1.136 MAX(Bereich[:])	47
1.137 MIN(Bereich[:])	48
1.138 MITTELWERT(Bereich[:])	48
1.139 PRODUKT(Bereich[:])	48
1.140 STABW(Bereich[:])	49
1.141 SUMME(Bereich[:])	49
1.142 VANZAHL(Bereich[:])	49
1.143 VANZAHL2(Bereich[:])	49
1.144 VARIANZ(Bereich[:])	50
1.145 VERSTER(Bereich;...)	50
1.146 VLETZTER(Bereich;...)	50

1.147VMAX(Bereich[;])	50
1.148VMIN(Bereich[;])	51
1.149VMITTELWERT(Bereich[;])	51
1.150VPRODUKT(Bereich[;])	51
1.151VSTABW(Bereich[;])	51
1.152VSUMME(Bereich[;])	52
1.153VVARIANZ(Bereich[;])	52
1.154VWÄHLEN(Index;Bereich;...)	52
1.155WÄHLEN(Index;Bereich;...)	52
1.156Datenbankfunktionen	53
1.157DBANZAHL(Datenbank;Spalte;Kriterien)	54
1.158DBANZAHL2(Datenbank;Spalte;Kriterien)	54
1.159DBERSTER(Datenbank;Spalte;Kriterien)	54
1.160DBLETZTER(Datenbank;Spalte;Kriterien)	55
1.161DBMAX(Datenbank;Spalte;Kriterien)	55
1.162DBMIN(Datenbank;Spalte;Kriterien)	55
1.163DBMITTELWERT(Datenbank;Spalte;Kriterien)	56
1.164DBPRODUKT(Datenbank;Spalte;Kriterien)	56
1.165DBSTABW(Datenbank;Spalte;Kriterien)	56
1.166DBSUMME(Datenbank;Spalte;Kriterien)	57
1.167DBVARIANZ(Datenbank;Spalte;Kriterien)	57
1.168DBWÄHLEN(Index;Datenbank;Spalte;Kriterien)	57
1.169Zellfunktionen	57
1.170#Bezug	58
1.171 @Tabelle;Bezug	59
1.172AKTUELLERBEREICH()	61
1.173AUSWAHL(Index; Wert1; Wert2; Wert3)	61
1.174BEREICHABS(Zeile;Spalte;Höhe;Breite)	61
1.175BLATTNAME()	62
1.176BLOCKBREITE([Bereich])	62
1.177BLOCKHÖHE([Bereich])	62
1.178BLOCKX([Bereich])	63
1.179BLOCKY([Bereich])	63
1.180INDIREKT(Text)	64
1.181MAPPENNAME()	64
1.182SPALTENNUMMER([Bereich])	64
1.183SVERWEIS(Wert;Bereich;Offset[;Exakt])	65
1.184TABELLENNAME()	65
1.185VERWEIS(Wert;Bereich[;Exakt])	66

1.186 WVERWEIS(Wert;Bereich;Offset[;Exakt])	66
1.187 ZEILENNUMMER([Bereich])	67
1.188 ZELLE(Zeile;Spalte)	67
1.189 ZELLEABS(Zeile;Spalte)	68
1.190 Finanzfunktionen	68
1.191 ENDKAPITAL(Kapital;Zinssatz;Zeitraum[;Periode])	69
1.192 LAUFZEIT(Kapital;Endwert;Zinssatz[;Periode])	69
1.193 RATENENDKAPITAL(Ratenhöhe;Zinssatz;Zeitraum[;Periode])	69
1.194 RATENHÖHE(Endwert;Zinssatz;Zeitraum[;Periode])	70
1.195 RATENLAUFZEIT(Entwert;Ratenhöhe;Zinssatz[;Periode])	70
1.196 STARTKAPITAL(Endwert;Zinssatz;Zeitraum[;Periode])	70
1.197 ZINSSATZ(Kapital;Endwert;Zeitraum[;Periode])	71
1.198 Sonstige Funktionen	71
1.199 ANIMBILD()	71
1.200 DATEIVORHANDEN(Datei)	72
1.201 DEMOVERSION()	72
1.202 INFO(Nr)	72
1.203 LETZTERFEHLER()	72
1.204 OBJEKTINFO(Name;Nr)	73
1.205 OBJID(Text)	73
1.206 REVISION()	73
1.207 SETZExxx(Bedingung;Wert1;Wert2[;Bezug])	73
1.208 TCFKT(TCLib;Offset;Num1;Num2;Text)	74
1.209 VERSION()	75
1.210 ZELLINFO(Nr[;Zelle])	75
1.211 Inhaltsverzeichnis	75
1.212 Index	81

Chapter 1

Funktionen

1.1 TurboCalc by Michael Friedrich

TurboCalc 5.0 - Benutzerhandbuch

copyright Michael Friedrich.

[Komplettes Inhaltsverzeichnis](#) dieser Datei.

Haupt-Inhaltsverzeichnis .

[Index](#) dieser Datei.

Kompletter Index (über alle Dateien).

Funktionen

Formeln bestehen neben Operatoren und normalen Werten hauptsächlich aus Funktionen, die auf den folgenden Seiten nach Gebieten getrennt beschrieben werden.

[Datentypen](#)

[Operatoren](#)

[Mathematikfunktionen](#)

[Wahrheitswert-Funktionen](#)

[Text-Funktionen](#)

[Datum- und Zeit-Funktionen](#)

[Tabellenfunktionen](#)

[Datenbankfunktionen](#)

[Zellfunktionen](#)

[Finanzfunktionen](#)

[Sonstige Funktionen](#)

1.2 Datentypen

Datentypen

Hier zunächst noch eine Übersicht über alle gängigen "Werte", wie sie von den Funktionen als Eingabeparameter (oder auch als Ausgabewert) benötigt werden.

Typ Beschreibung

Zahlen: Dies sind Zahlen im üblichen Format (+/-123[.123][E+-1]). Eine nähere Beschreibung finden Sie im Abschnitt "Eingabe".

Wahrheitswerte: Dies sind WAHR oder FALSCH. Statt normalen Wahrheitswerten können auch Zahlen benutzt werden. (0 bedeutet dann FALSCH, alles andere WAHR).

Texte: Diese müssen in Anführungszeichen stehen und können beliebige Zeichen enthalten. Liefert eine Formel als Ergebnis einen Text, so wird dieser natürlich ohne Anführungszeichen angezeigt.

Tip: Soll in einem Text das Anführungszeichen benutzt werden, so müssen für jedes Anführungszeichen zwei aufeinanderfolgende Anführungszeichen geschrieben werden ("a""b" ist der Text a"b)

Datum: Dies ist eine fortlaufende Ganzzahl (0 bedeutet 1.1.1900, 1: 2.1.1900...) die jedoch nicht als Zahl sondern als richtiges Datum eingegeben werden sollte. Hierzu muß jedoch, im Gegensatz zur Eingabe eines Datums in eine Zelle, die Funktion DATWERT("Text") oder einfach WERT("Text") benutzt werden. Siehe auch "implizite Typumwandlung" unten. Beispiel: WERT("1-9-93").

Uhrzeit: Dies ist eine fortlaufende Ganzzahl (in Sekundenschritten) (0 bedeutet 0 Uhr, 1: 0:00:01...) die jedoch nicht als Zahl sondern als richtige Uhrzeit eingegeben werden sollte. Hierzu muß jedoch, im Gegensatz zur Eingabe in eine Zelle, die Funktion ZEITWERT("Text") oder einfach WERT("Text") benutzt werden. Siehe auch "implizite Typumwandlung" unten. Beispiel: WERT("12:03:07").

Bezug: Hiermit kann man Inhalte von Zellen auslesen und den Inhalt als Wert benutzen. Bezüge bestehen aus einem (bis drei) Buchstaben gefolgt von einer Zahl (ohne Leerzeichen!), etwa A1 oder CC23. Die Buchstaben geben dabei die Spalte und die Zahl die Zeile an. Steht in der Zelle C7 etwa 123, so ergibt "=C7" die Zahl 123.

Bezüge können auch absolut angegeben werden (d.h. sie werden beim Kopieren, Verschieben, usw. nicht verändert), indem man vor die jeweilige "feste" Richtung (oder beide) ein Dollarzeichen stellt, etwa \$C\$7 oder \$C7 oder C\$7.

Möchte man Werte aus anderen Tabellen auslesen, so kann man dazu die Funktion @ (oder AT) benutzen, siehe bei Zell-Funktionen.

Bereich: Dies sind Parameter für die Bereichs- und die Datenbankfunktionen. Bereiche bestehen aus zwei durch Doppelpunkt getrennten Bezügen (etwa A1:C5 oder \$C3: \$E5) und bestimmen das Rechteck zwischen diesen beiden Zellen. (Ein Bereich kann auch aus nur einem Bezug bestehen, dann ist damit der 1 mal 1 große Bereich des Bezugs gemeint).

Namen: Für alle obigen Typen können auch Namen (d.h. Variablen) benutzt werden. Einzelheiten dazu siehe im Abschnitt "Namen".

implizite Typumwandlung

Zur Typumwandlung können die Funktionen WERT, DATWERT sowie ZEITWERT (um einen Text in eine Zahl, ein Datum oder aber eine Zeit umzuwandeln) sowie TEXT (für die umgekehrte Richtung) benutzt werden. WERT wandelt dabei - analog zur Zelleingabe - automatisch in den passenden Typ (Zahl, Datum, Zeit oder Wahrheitswert) um.

Ab der Version 4 wird dies bei Funktionen auch automatisch durchgeführt, falls Sie als Parameter einen Text angeben (bzw. einen Verweis auf eine Zelle, die einen Text enthält) und der benötigte Typ des Parameters kein Text sein kann.

Dies eignet sich insbesondere für die Typen Datum und Zeit, die ja nicht direkt als Parameter angegeben werden können.

Hinweise:

Dies ist nur bei Funktionen, nicht jedoch bei Makrobefehlen oder aber Operatoren möglich.

Bei den Tabellenfunktionen (MIN, MAX, SUMME, ...) werden laut Beschreibung Texte ignoriert. Entsprechend findet bei diesen Funktionen keine implizite Typumwandlung statt.

Beispiele:

SIN("0.5")

WOCHENTAG("1.4.1996")

MINUTEN("13:05")

1.3 Operatoren

Operatoren

Operatoren stehen, im Gegensatz zu den unten aufgeführten Funktionen, zwischen zwei Werten und bestimmen, wie diese verknüpft werden sollen.

Hier eine Übersicht über alle möglichen Operatoren.

Pri. Operator Beschreibung

5 ^ (hoch) potenziert

4 * multipliziert

4 / dividiert

4 REST (MOD) Berechnet den Modulo, siehe bei der Mathematikfunktion REST

3 + addiert (auch Texte, Uhrzeiten und Datum, siehe bei den entsprechenden Funktionsbeschreibungen)

3 - subtrahiert (auch Texte, etc. siehe bei "+")

2 =, >, <, <>, >=, <= Vergleicht die beiden Werte und gibt entweder WAHR oder FALSCH zurück (auch Texte, Uhrzeiten und Daten).

1 &, UND, AND Verknüpft die Werte mit logischem UND. Siehe bei Wahrheitswertfunktionen

1 |, ODER, OR Verknüpft die Werte mit logischem ODER. Siehe bei Wahrheitswertfunktionen

1 XOR, EOR Verknüpft die Werte mit logischen XOR. Siehe bei Wahrheitswertfunktionen unter XOR.

Die Priorität (Pri.) gibt dabei an, in welcher Reihenfolge die Berechnung stattfinden soll, falls keine Klammern gesetzt sind. Dann werden zuerst die Operationen mit der höchsten Priorität ausgeführt, danach die anderen. Haben Operatoren die gleiche Priorität, so wird von links nach rechts gerechnet. Dies entspricht der Standard-Regel "Punkt vor Strich".

Beispiele:

$2+2*3^2+2$ wird als $(2+(2*(3^2)))+2$ berechnet.

$2-2-2$ wird als $(2-2)-2$ berechnet (bei gleicher Priorität von links nach rechts) und ergibt -2, obwohl natürlich auch $2-(2-2) = 2$ möglich wäre.

1.4 Mathematikfunktionen

Mathematikfunktionen

Dies sind die normalen mathematischen Funktionen. Die meisten davon werden Sie auf Taschenrechnern oder in Programmiersprachen finden.

Zahl: Falls nicht anders bezeichnet, handelt es sich beim Eingabeparameter "Zahl" um eine normale reelle Zahl.

ABRUNDEN (Zahl;Stellen)

ABS(Zahl)

ARCCOS(Zahl)

ARCSIN(Zahl)

ARCTAN(Zahl)

AUFRUNDEN(Zahl;Stellen)

BOGEN(Zahl)

COS(Zahl)

COSHYP(Zahl)

EXP(Zahl)
EUKLID1 (Zahl1;Zahl2)
EUKLID2 (Zahl1;Zahl2)
FAKULTÄT(Zahl)
GANZZAHL(Zahl)
GERADE(Zahl)
GGT(Zahl1;Zahl2)
GRENZE(X;a;b;W1;W2;W3)
HOCH(Basis;Exponent)
HOCH10(Exponent)
KEHRWERT(Zahl)
KGV(Zahl1;Zahl2)
KOMBINATIONEN(n;k)
KUM([Zahl1;]Zahl2)
KÜRZEN(Zahl)
LG(Zahl)
LN(Zahl)
LOG(Zahl)
LOG10(Zahl)
MINVERS(Zahl;Modulo)
NACHKOMMA(Zahl)
OBERGRENZE(Zahl;Schritt)
PI()
QUADRAT(Zahl)
REST(Zahl1;Zahl2)
RUNDEN(Zahl;Stellen)
SIN(Zahl)
SINHYP(Zahl)
TAN(Zahl)
TANHYP(Zahl)
UNGERADE(Zahl)
UNTERGRENZE(Zahl;Schritt)
VARIATIONEN(n;k)
VORZEICHEN(Zahl)
WINKEL(Zahl)
WURZEL(Zahl[;Basis])
ZUFALLSBEREICH(Min;Max)
ZUFALLSZAHL()
ZWEIFAKULTÄT(Zahl)

1.5 ABRUNDEN (Zahl;Stellen)

ABRUNDEN (Zahl;Stellen)

(Englisch: ROUNDDOWN)

Diese Funktion rundet eine Zahl auf eine bestimmte Dezimalstelle ab. Sie unterscheidet sich von der Funktion RUNDEN dadurch, daß sie eine Zahl immer abrundet.

Zahl ist die abzurundende reelle Zahl.

Stellen bestimmt, auf wie viele Dezimalstellen gerundet werden soll. (Wird dieser Parameter ausgelassen, so wird 0 benutzt, d.h. auf die ganze Zahl gerundet)

Stellen>0: rundet auf Stellen Dezimalstellen nach dem Komma.

Stellen<0: rundet auf die entsprechenden Dezimalstellen links vom Komma (also 10,100,1000...)

Stellen=0: rundet auf die ganze Zahl

Beispiele:

ABRUNDEN(3.19;1) = 3.1

ABRUNDEN(314.5;-2) = 300

Verwandte Funktionen:

AUFRUNDEN, RUNDEN, GANZZAHL

1.6 ABS(Zahl)

ABS(Zahl)

Die Funktion ABS liefert als Ergebnis den Absolutwert (d.h. den Betrag) einer Zahl, d.h. die Zahl ohne Vorzeichen.

Beispiele:

ABS(2) ergibt 2

ABS(-2) ergibt 2

Verwandte Funktionen:

VORZEICHEN

1.7 ARCCOS(Zahl)

ARCCOS(Zahl)

Die Funktion ARCCOS liefert den Arcuscosinus einer Zahl zurück. Dies ist der Winkel dessen Kosinus (COS) Zahl ergibt. Der Ergebniswinkel wird im Bogenmaß im Wertebereich 0 bis PI angegeben. (Zur Umrechnung ins Gradmaß: siehe WINKEL)

Zahl ist der Kosinus des gesuchten Winkels und muß zwischen -1 und 1 liegen (sonst wird der Fehler #WERT ausgegeben).

Beispiele:

ARCCOS(-0.5) ergibt 2.094

WINKEL(ARCCOS(-0.5)) ergibt 120 (Grad)

Verwandte Funktionen:

COS, PI, WINKEL

1.8 ARCSIN(Zahl)

ARCSIN(Zahl)

Die Funktion ARCSIN liefert den Arcussinus einer Zahl zurück. Dies ist der Winkel dessen Sinus (SIN) Zahl ergibt. Der Ergebniswinkel wird im Bogenmaß im Wertebereich $-\pi/2$ bis $\pi/2$ angegeben. (Zur Umrechnung ins Gradmaß: siehe WINKEL)

Zahl ist der Sinus des gesuchten Winkels und muß zwischen -1 und 1 liegen (sonst wird der Fehler #WERT ausgegeben).

Beispiele:

ARCSIN(-0.5) ergibt 0.524

WINKEL(ARCSIN(-0.5)) ergibt -30 (Grad)

Verwandte Funktionen:

SIN, PI, WINKEL

1.9 ARCTAN(Zahl)

ARCTAN(Zahl)

Die Funktion ARCTAN liefert den Arkustangens einer Zahl zurück. Dies ist der Winkel dessen Tangens (TAN) Zahl ergibt. Der Ergebniswinkel wird im Bogenmaß im Wertebereich $-\pi/2$ bis $\pi/2$ angegeben. (Zur Umrechnung ins Gradmaß: siehe WINKEL)

Beispiele:

ARCTAN(1) ergibt 0.785

WINKEL(ARCTAN(-1)) ergibt 45 (Grad)

Verwandte Funktionen:

TAN, PI, WINKEL

1.10 AUFRUNDEN(Zahl;Stellen)

AUFRUNDEN(Zahl;Stellen)

(Englisch: ROUNDUP)

Diese Funktion rundet eine Zahl auf eine bestimmte Dezimalstelle auf. Sie unterscheidet sich von der Funktion RUNDEN dadurch, daß sie eine Zahl immer aufrundet.

Zahl ist die zu rundende reelle Zahl.

Stellen bestimmt, auf wie viele Dezimalstellen gerundet werden soll. (Wird dieser Parameter ausgelassen, so wird 0 benutzt, d.h. auf die ganze Zahl gerundet)

Stellen>0: rundet auf Stellen Dezimalstellen nach dem Komma.

Stellen<0: rundet auf die entsprechenden Dezimalstellen links vom Komma (also 10,100,1000...)

Stellen=0: rundet auf die ganze Zahl

Beispiele:

AUFRUNDEN(3.14;1) = 3.2

AUFRUNDEN(314.5;-2) = 400

Verwandte Funktionen:

ABRUNDEN, RUNDEN, GANZZAHL

1.11 BOGEN(Zahl)

BOGEN(Zahl)

(Englisch: DEGTORAD)

Die Funktion BOGEN wandelt einen Winkel vom Gradmaß in das Bogenmaß um.

Damit können normale Winkelangaben als Parameter für die Funktionen Sinus, Cosinus und Tangens benutzt werden.

Beispiele:

$\text{BOGEN}(30) = 0.523 (= \pi/6)$

$\text{SIN}(\text{BOGEN}(30)) = 0.5$

Verwandte Funktionen:

WINKEL, SIN, COS, TAN

1.12 COS(Zahl)

COS(Zahl)

Die Funktion COS liefert den Kosinus des Winkels Zahl (im Bogenmaß). Zur Umrechnung von Winkeln ins Bogenmaß siehe BOGEN.

Beispiele:

$\text{COS}(1.047)$ ergibt 0.5

$\text{COS}(\text{BOGEN}(60))$ ergibt 0.5

Verwandte Funktionen:

ARCCOS, PI, BOGEN

1.13 COSHYP(Zahl)

COSHYP(Zahl)

Die Funktion COSHYP liefert den Cosinus hyperbolicus von Zahl (d.h. $=1/2*(e^z+e^{-z})$)

Beispiele:

$\text{COSHYP}(2)$ ergibt 3.762

Verwandte Funktionen:

EXP, SINHYP, TANHYP

1.14 EXP(Zahl)

EXP(Zahl)

Die Funktion EXP liefert als Ergebnis den Wert e ($=2.718281828$) hoch Zahl.

Beispiele:

$\text{EXP}(1)$ ergibt 2.71828...

$\text{EXP}(\text{LN}(5))$ ergibt 5

Verwandte Funktionen:

LN, LOG, LOG10, LG

1.15 EUKLID1 (Zahl1;Zahl2)

EUKLID1 (Zahl1;Zahl2)

(Englisch: EUCLID1)

Diese Funktionen (EUKLID1 und EUKLID2) liefern zwei Zahlen i und j für die folgende Beziehung gilt:

$$i * \text{Zahl1} + j * \text{Zahl2} = \text{GGT}(\text{Zahl1}; \text{Zahl2})$$

Beispiele:

$$\text{GGT}(228;612) = 12$$

$$\text{EUCLID1}(228;612) = -8$$

$$\text{EUCLID2}(228;612) = 3$$

$$\text{Probe: } -8 * 228 + 3 * 612 = 12$$

Verwandte Funktionen:

GGT, **KGV**

1.16 EUKLID2 (Zahl1;Zahl2)

EUKLID2 (Zahl1;Zahl2)

(Englisch: EUCLID2)

Siehe EUKLID1.

1.17 FAKULTÄT(Zahl)

FAKULTÄT(Zahl)

(Englisch: FAC)

Die Funktion FAKULTÄT liefert als Ergebnis die Fakultät einer Zahl, d.h. $1 * 2 * 3 * \dots * \text{Zahl}$

Zahl sollte daher positiv und ganz sein. (Kommazahlen werden abgeschnitten)

Hinweis: Statt FAKULTÄT(3) kann auch einfach $3!$ geschrieben werden.

Beispiele:

FAKULTÄT(5) ergibt 120

FAKULTÄT(5.2) ergibt 120

$5!$ ergibt auch 120

1.18 GANZZAHL(Zahl)

GANZZAHL(Zahl)

(Englisch: INTEGER, oder auch kurz GANZ bzw. INT)

Die Funktion GANZZAHL rundet Zahl auf die nächste ganze Zahl ab.

Beispiele:

GANZZAHL(5.4) ergibt 5

GANZZAHL(5.7) ergibt 5

GANZZAHL(-5.2) ergibt -6

Verwandte Funktionen:

REST, **RUNDEN**, **KÜRZEN**

1.19 GERADE(Zahl)

GERADE(Zahl)

(Englisch: EVEN)

Diese Funktion rundet eine Zahl auf die nächste gerade ganze Zahl.

Beispiele:

GERADE(5) ergibt 6

GERADE(4.1) ergibt 4

Verwandte Funktionen:

UNGERADE, ISTGERADE, ISTUNGERADE

1.20 GGT(Zahl1;Zahl2)

GGT(Zahl1;Zahl2)

(Englisch: GCD)

Diese Funktion liefert den größten gemeinsamen Teiler zweier ganzer Zahlen. Der größte gemeinsame Teiler ist die ganze Zahl, durch die sowohl Zahl1 als auch Zahl2 ohne Rest dividiert werden können.

Zahl1, Zahl2: Ganze Zahlen von denen der GGT ermittelt werden soll. Bei Werten, die keine ganzen Zahlen sind, werden deren Nachkommastellen abgeschnitten.

Beispiele:

GGT(5;2) ergibt 1

GGT(24;36) ergibt 12

Verwandte Funktionen:

KGV, EUKLID1, EUKLID2

1.21 GRENZE(X;a;b;W1;W2;W3)

GRENZE(X;a;b;W1;W2;W3)

(Englisch: LIMIT)

Diese Funktion teilt den Zahlenstrahl in drei Teilbereiche (...a...b...) auf und gibt, in Abhängigkeit von X einen der drei Werte W1,W2 oder W3 zurück. Es ist somit ein erweitertes (doppeltes) WENN.

X Wert

$X < a$: W1

$a \leq X \leq b$: W2

$X > b$: W3

Beispiele:

LIMIT(A1;0;2;"Fehler1";"OK";"Fehler2") gibt, in Abhängigkeit vom Inhalt der Zelle A1 einen der drei Texte aus. "OK" wird hierbei ausgegeben, falls A1 positiv (größer oder gleich 0) aber nicht größer als 2 ist.

Kosten in Abhängigkeit von der Stückzahl A1 berechnen: Bei weniger als 10 Stücken betrage der Stückpreis 1DM, 10 bis 99 Exemplare kosten je 50Pf und ab 100 Exemplaren sei der Stückpreis 20Pf. Damit berechnen sich die Gesamtkosten wie folgt:

LIMIT(A1;A1;99;10;A1*0,5;A1*0,20)

Verwandte Funktionen:

WENN, AUSWAHL

1.22 HOCH(Basis;Exponent)

HOCH(Basis;Exponent)

(Englisch: POW)

Diese Funktion berechnet $\text{Basis}^{\text{Exponent}}$. Statt dieser Funktion kann auch der Operator "^" verwandt werden.

Verwandte Funktionen:

HOCH10, EXP, LN, LOG10

1.23 HOCH10(Exponent)

HOCH10(Exponent)

(Englisch: POW10)

Diese Funktion berechnet 10^{Exponent} . Es ist die Umkehrfunktion zu LOG10. Statt dieser Funktion kann auch der Operator "10^" verwandt werden.

Verwandte Funktionen:

HOCH, EXP, LN, LOG10

1.24 KEHRWERT(Zahl)

KEHRWERT(Zahl)

(Englisch: RECIPROCAL)

Diese Funktion berechnet das multiplikative Inverse (Kehrwert) einer Zahl. Statt dieser Funktion kann auch der Operator "1/" verwandt werden.

Verwandte Funktionen:

MINVERS

1.25 KGV(Zahl1;Zahl2)

KGV(Zahl1;Zahl2)

(Englisch: LCM)

Diese Funktion liefert das kleinste gemeinsame Vielfache zweier ganzer Zahlen. Als kleinstes gemeinsames Vielfaches wird die kleinste positive ganze Zahl bezeichnet, die ein Vielfaches aller ganzzahligen Argumente Zahl1, Zahl2 und so weiter ist. KGV können Sie verwenden, wenn Sie Brüche addieren müssen, die unterschiedliche Nenner haben.

Beispiele:

KGV(5;2) ergibt 10

KGV(24;36) ergibt 72

Verwandte Funktionen:

GGT, EUKLID1, EUKLID2

1.26 KOMBINATIONEN(n;k)

KOMBINATIONEN(n;k)

(Englisch: COMB)

Liefert die Anzahl der Kombinationen ohne Wiederholung von k Elementen aus einer Menge von n Elementen. Verwenden Sie KOMBINATIONEN, wenn Sie berechnen möchten, wie viele Gruppen aus einer bestimmten Anzahl von Elementen gebildet werden können.

Eine Kombination ist eine beliebige Menge oder Teilmenge von Elementen, deren interne Reihenfolge keine Rolle spielt. Kombinationen unterscheiden sich von Permutationen, bei denen die interne Reihenfolge der ausgewählten Elementen von Bedeutung ist.

n ist die Anzahl aller Elemente. ($n \geq 0$)

k gibt an, aus wie vielen Elementen jede Kombination bestehen soll. ($k \geq 0$, $k \leq n$)

Beispiel:

KOMBINATIONEN(9;3) ergibt 84

Verwandte Funktionen:

VARIATIONEN

1.27 KUM([Zahl1;]Zahl2)

KUM([Zahl1;]Zahl2)

(Englisch: CUM)

Dies summiert alle ganzen Zahlen im Bereich von Zahl1 bis Zahl2 auf. Wird Zahl1 ausgelassen, so wird 1 angenommen und alle Zahlen von 1 bis Zahl2 aufsummiert.

Zahl1, Zahl2: Ganze Zahlen größer oder gleich Null. Ist eine der beiden Zahlen negativ, so wird der Fehlerwert #WERT zurückgegeben.

Beispiele:

CUM(10) ergibt $1+2+\dots+9+10=55$

CUM(6;10) ergibt $6+7+8+9+10=40$

Verwandte Funktionen:

FAKULTÄT

1.28 KÜRZEN(Zahl)

KÜRZEN(Zahl)

(Englisch: TRUNC)

Die Funktion KÜRZEN rundet Zahl auf die nächste ganze Zahl, indem die Nachkommastellen abgeschnitten werden.

Im Gegensatz zu **GANZZAHL** schneidet KÜRZEN die Nachkommastellen ab und rundet nicht auf die nächste kleiner ganze Zahl (Im dritten Beispiel würde sich bei GANZZAHL(-5.2) der Wert -6 ergeben).

Beispiele:

KÜRZEN(5.4) ergibt 5

KÜRZEN(5.7) ergibt 5

KÜRZEN(-5.2) ergibt -5

Verwandte Funktionen:

REST, RUNDEN, GANZZAHL

1.29 LG(Zahl)

LG(Zahl)

Synonym zu LOG10 (siehe unten) welches bei Mathematikern sehr geläufig ist.

1.30 LN(Zahl)

LN(Zahl)

Die Funktion LN liefert den natürlichen Logarithmus einer Zahl (d.h. zur Basis $e=2.718281828\dots$)

Zahl ist eine positive reelle Zahl (d.h. beliebige Zahl >0). Ansonsten erscheint der Fehlerwert #WERT.

LN ist die Umkehrfunktion zu EXP

Beispiele:

$$\text{LN}(100) = 4.605$$

$$\text{LN}(\text{EXP}(3)) = 3$$

Verwandte Funktionen:

EXP, LOG, LOG10

1.31 LOG(Zahl)

LOG(Zahl)

Synonym zu LN (siehe oben) welches bei Mathematikern sehr geläufig ist.

1.32 LOG10(Zahl)

LOG10(Zahl)

Die Funktion LOG10 liefert den Logarithmus einer Zahl zur Basis 10

Zahl ist eine positive reelle Zahl.

Beispiele:

$$\text{LN}(100) = 2$$

$$\text{LN}(10^5) = 5$$

Verwandte Funktionen:

EXP, LN, LG, LOG, LOG10

1.33 MINVERS(Zahl;Modulo)

MINVERS(Zahl;Modulo)

(Englisch: MINVERSE)

Diese Funktion berechnet das Inverse von Zahl im Restklassenring \mathbb{Z}_n (d.h. bei der Multiplikation modulo n).

$$\text{MINVERS}(\text{Zahl};\text{Modulo}) * \text{Zahl} \bmod \text{Modulo} = 1$$

Hinweis: Ist Modulo eine Primzahl so ist der Ring nullteilerfrei und somit ein Körper. In diesem Fall existiert zu jeder Zahl (1...Modulo-1) ein Inverses. Ansonsten existieren Nullteiler für die es kein Inverses modulo Modulo gibt; in diesem Fall liefert die Funktion den Fehlerwert #WERT zurück.

Beispiele:

MINVERS(24;59) ergibt -27 ($=32 \bmod 59$)

Probe: $-27 * 24 \bmod 59 = 1$

MINVERS(2;6) ergibt #WERT

6 ist keine Primzahl und 2 ist Nullteiler ($2*3=6$), somit existiert kein Inverses.

Verwandte Funktionen:

REST

1.34 NACHKOMMA(Zahl)

NACHKOMMA(Zahl)

(Englisch: FRAC)

Diese Funktion liefert die Nachkommastellen einer Zahl

Beispiele:

NACHKOMMA(1,234) ergibt 0,234

Verwandte Funktionen:

GANZZAHL

1.35 OBERGRENZE(Zahl;Schritt)

OBERGRENZE(Zahl;Schritt)

(Englisch: CEILING)

Diese Funktion rundet eine Zahl auf das nächste Vielfache von Schritt auf.

Zahl ist der Wert, den Sie runden möchten.

Schritt ist der Wert, auf dessen Vielfaches Sie aufrunden möchten. Ist Zahl ein genaues Vielfaches von Schritt, erfolgt keine Rundung.

Beispiele:

Wenn Sie beispielsweise verhindern möchten, daß bei Ihren Preisen Pfennige verwendet werden, wobei Ihr Produkt DM 3,13 kostet, können Sie die folgende Formel verwenden, um die Preise entsprechend einer 5-Pfennig-Stufung aufzurunden:

OBERGRENZE(3,13;0,05) ergibt 3,15

OBERGRENZE(-3,13;0,05) ergibt -3,10

OBERGRENZE(-3,13;-0,05) ergibt -3,15

OBERGRENZE(5;4) ergibt 8

Verwandte Funktionen:

UNTERGRENZE, AUFRUNDEN, ABRUNDEN, RUNDEN, GANZZAHL

1.36 PI()

PI()

Die Funktion PI liefert die Zahl 3.1415926...

Beispiele:

$$\text{PI()}/2 = 1.5707\dots$$

$$\text{PI}/2 = 1.5707\dots$$

$$\text{SIN}(\text{PI}/2) = 1$$

Dies wird zur Berechnung des Umfangs und der Fläche von Kreisen/Kugeln benutzt. Steht in A1 der Radius eines Kreises, so ergibt:

$2 \cdot \text{PI} \cdot \text{A1}$ den Umfang und

$\text{PI} \cdot (\text{A1}^2)$ die Fläche des Kreises.

1.37 QUADRAT(Zahl)

QUADRAT(Zahl)

(Englisch: SQR)

Die Funktion QUADRAT liefert das Quadrat von Zahl, also $\text{Zahl} \cdot \text{Zahl}$.

Beispiele:

$$\text{QUADRAT}(5) = 25$$

$$5^2 = 25$$

Verwandte Funktionen:

WURZEL

1.38 REST(Zahl1;Zahl2)

REST(Zahl1;Zahl2)

(Englisch: MOD)

Die Funktion REST bestimmt den Rest bei der Division von Zahl1 durch Zahl2.

Zahl1 und Zahl2 können dabei beliebige reelle Zahlen sein (Zahl2 ungleich 0!)

Beispiele:

$$\text{REST}(10;3) \text{ ergibt } 1$$

$$\text{REST}(10.5;3) \text{ ergibt } 1.5$$

$$\text{REST}(2.25;0.5) \text{ ergibt } 0.25$$

1.39 RUNDEN(Zahl;Stellen)

RUNDEN(Zahl;Stellen)

(Englisch: ROUND)

Die Funktion RUNDEN rundet eine Zahl auf eine bestimmte Dezimalstelle.

Zahl ist die zu rundende reelle Zahl.

Stellen bestimmt, auf wie viele Dezimalstellen gerundet werden soll. (Wird dieser Parameter ausgelassen, so wird 0 benutzt, d.h. auf die ganze Zahl gerundet)

Stellen>0: rundet auf "Stellen" Stellen nach dem Komma.

Stellen<0: rundet auf die entsprechenden Stellen links vom Komma (also 10,100,1000...)

Stellen=0: rundet auf die ganze Zahl

Beispiele:

$\text{RUNDEN}(3.14;1) = 3.1$

$\text{RUNDEN}(3.15;1) = 3.2$

$\text{RUNDEN}(314.5;-2) = 300$

Verwandte Funktionen:

AUFRUNDEN, ABRUNDEN, GANZZAHL

1.40 SIN(Zahl)

SIN(Zahl)

Die Funktion SIN liefert den Sinus des Winkels Zahl (im Bogenmaß). Zur Umrechnung von Winkeln ins Bogenmaß siehe BOGEN.

Beispiele:

$\text{SIN}(1.047)$ ergibt 0.8659

$\text{SIN}(\text{BOGEN}(60))$ ergibt 0.8659

Verwandte Funktionen:

ARCSIN, PI, BOGEN

1.41 SINHYP(Zahl)

SINHYP(Zahl)

Die Funktion SINHYP liefert den Sinus hyperbolicus von Zahl (d.h. $=1/2*(e^z - e^{-z})$)

Beispiele:

$\text{SINHYP}(2)$ ergibt 3.627

Verwandte Funktionen:

EXP, COSHYP, TANHYP

1.42 TAN(Zahl)

TAN(Zahl)

Die Funktion TAN liefert den Tangens des Winkels Zahl (im Bogenmaß) (d.h. SIN/COS). Zur Umrechnung von Winkeln ins Bogenmaß siehe BOGEN.

Beispiele:

$\text{TAN}(0.7854)$ ergibt 1

$\text{TAN}(\text{BOGEN}(45))$ ergibt 1

Verwandte Funktionen:

ARCTAN, COS, PI, BOGEN, SIN

1.43 TANHYP(Zahl)

TANHYP(Zahl)

Die Funktion TANHYP liefert den Tangens hyperbolicus von Zahl (d.h. $=\text{SINHYP(Zahl)}/\text{COSHYP(Zahl)}$)

Beispiele:

TANHYP(2) ergibt 0.964

Verwandte Funktionen:

EXP, SINHYP, COSHYP

1.44 UNGERADE(Zahl)

UNGERADE(Zahl)

(Englisch: ODD)

Diese Funktion rundet eine Zahl auf die nächste ungerade ganze Zahl.

Beispiele:

UNGERADE(6) ergibt 7

UNGERADE(5.1) ergibt 5

Verwandte Funktionen:

GERADE, ISTGERADE, ISTUNGERADE

1.45 UNTERGRENZE(Zahl;Schritt)

UNTERGRENZE(Zahl;Schritt)

(Englisch: FLOOR)

Diese Funktion rundet eine Zahl auf das nächste Vielfache von Schritt ab.

Zahl ist der Wert, den Sie runden möchten.

Schritt ist der Wert, auf dessen Vielfaches Sie abrunden möchten. Ist Zahl ein genaues Vielfaches von Schritt, erfolgt keine Rundung.

Beispiele:

UNTERGRENZE(3,13;0,05) ergibt 3,10

UNTERGRENZE(-3,13;0,05) ergibt -3,15

UNTERGRENZE(-3,13;-0,05) ergibt -3,10

UNTERGRENZE(5;4) ergibt 4

Verwandte Funktionen:

OBERGRENZE, AUFRUNDEN, ABRUNDEN, RUNDEN, GANZZAHL

1.46 VARIATIONEN(n;k)

VARIATIONEN(n;k)

(Englisch: PERM)

Liefert die Anzahl der Möglichkeiten, um k Elemente aus einer Menge von n Elementen ohne Zurücklegen zu ziehen. Eine Variation ist eine Kombination ohne Wiederholung, deren interne Anordnung oder Reihenfolge zu berücksichtigen ist (z.B. Pferdewetten). Variationen unterscheiden sich von Kombinationen, deren interne Anordnung unberücksichtigt bleibt (z.B. Zahlenlotto 6 aus 49).

n ist die Anzahl aller Elemente ($n > 0$).

k gibt an, aus wie vielen Elementen jede Variationsmöglichkeit bestehen soll. ($k \geq 0$ sowie $k \leq n$)

Beispiele:

In einer Schachtel befinden sich neun Zettel mit den Ziffern 1 bis 9. Sie ziehen drei diese Zettel und stellen so eine dreistellige Zahl dar (erster Zettel entspricht erster Ziffer der Zahl).

Wie viele verschiedene Zahlen gibt es:

VARIATIONEN(9;3) ergibt 504

Verwandte Funktionen:

KOMBINATIONEN

1.47 VORZEICHEN(Zahl)

VORZEICHEN(Zahl)

(Englisch: SIGN)

Die Funktion Vorzeichen liefert als Ergebnis -1,0,1 in Abhängigkeit vom Vorzeichen der Zahl:

-1 wenn $Zahl < 0$, 0 falls $Zahl = 0$ und 1 falls $Zahl > 0$.

Beispiele:

VORZEICHEN(20) = 1

VORZEICHEN(-20) = -1

VORZEICHEN(0) = 0

VORZEICHEN(A1)*ABS(A1) liefert gerade A1 zurück (falls A1 eine Zahl enthält)

Verwandte Funktionen:

ABS

1.48 WINKEL(Zahl)

WINKEL(Zahl)

(Englisch: RADTODEG)

Die Funktion WINKEL wandelt einen Winkel vom Bogenmaß in das Gradmaß um.

Damit können Winkelangaben der Funktionen ARCSIN, ARCCOS und ARCTAN in normale Winkel umgerechnet werden.

Beispiele:

WINKEL(0.5236) = 30

WINKEL(ARCSIN(0.5)) = 30

Verwandte Funktionen:

BOGEN, ARCSIN, ARCCOS, ARCTAN

1.49 WURZEL(Zahl[;Basis])

WURZEL(Zahl[;Basis])

(Englisch: SQRT)

Die Funktion WURZEL liefert die positive Quadratwurzel einer Zahl.

Zahl ist ein reelle positive Zahl.

Basis ist optional und bestimmt zu welcher Basis die Wurzel gezogen werden soll. Wird dieser Parameter ausgelassen, so wird automatisch 2 benutzt, d.h. die Quadratwurzel.

Beispiele:

WURZEL(25) = 5

WURZEL(QUADRAT(5)) = 5

WURZEL(-25) = #WERT

WURZEL(ABS(-25)) = 5

WURZEL(125;3) ergibt 5

Verwandte Funktionen:

QUADRAT, **ABS**

1.50 ZUFALLSBEREICH(Min;Max)

ZUFALLSBEREICH(Min;Max)

(Englisch: RNDRange)

Diese Funktion liefert eine Zufallszahl zwischen Min und Max. (Min einschließlich, Max ausschließlich)

Beispiele:

GANZZAHL(ZUFALLSBEREICH(1;7)) liefert eine zufällige ganze Zahl zwischen 1 und 6, simuliert also einen Würfel.

Verwandte Funktionen:

ZUFALLSZAHL

1.51 ZUFALLSZAHL()

ZUFALLSZAHL()

(Englisch: RND)

Die Funktion ZUFALLSZAHL liefert eine Zufallszahl zwischen 0 und 1. (0 einschließlich, 1 ausschließlich)

Will man eine Zufallszahl zwischen 1 und N, so kann dies wie folgt erreicht werden:

GANZZAHL(ZUFALLSZAHL()*N+1)

Beispiele:

GANZZAHL(ZUFALLSZAHL()*6+1) liefert eine zufällige Zahl zwischen 1 und 6, simuliert also einen Würfel.

Verwandte Funktionen:

ZUFALLSBEREICH

1.52 ZWEIFAKULTÄT(Zahl)

ZWEIFAKULTÄT(Zahl)

(Englisch: TWOFAC)

Liefert die Fakultät zu Zahl mit Schrittlänge 2.

Zahl: positive ganze Zahl, dessen Fakultät mit Schrittlänge zwei Sie berechnen möchten. Ist Zahl keine ganze Zahl, werden deren Nachkommastellen abgeschnitten.

Es gilt:

$ZWEIFAKULTÄT(n) = n \cdot (n-2) \cdot (n-4) \cdot \dots \cdot 2$, falls n gerade

$n \cdot (n-2) \cdot (n-4) \cdot \dots \cdot 1$, falls n ungerade

Beispiele:

ZWEIFAKULTÄT(6) ist gleich $2 \cdot 4 \cdot 6$ und ergibt 48

ZWEIFAKULTÄT(7) ist gleich $1 \cdot 3 \cdot 5 \cdot 7$ und ergibt 105

Verwandte Funktionen:

FAKULTÄT

1.53 Wahrheitswert-Funktionen

Wahrheitswert-Funktionen

Diese Funktionen liefern als Ergebnis einen Wahrheitswert (WAHR bzw. FALSCH, d.h. 1 bzw. 0) zurück bzw. verarbeiten solche Wahrheitswerte.

Diese Funktionen eignen sich damit gut für bedingte Berechnungen und Entscheidungen. (Siehe vor allem WENN.)

ÄHNLICH(Text1;Text2)

ENTHÄLT(Text1;Text2)

FALSCH()

IDENTISCH(Text1;Text2)

ISTDATUM(Wert)

ISTFEHLER(Bezug)

ISTGERADE(Zahl)

ISTLEER(Bezug)

ISTTEXT(Wert)

ISTUNGERADE(Zahl)

ISTZAHL(Wert)

ISTZEIT(Wert)

MUSTERVERGLEICH(Text;Muster)

NICHT(Wert)

ODER(Wert1;Wert2;)

UND(Wert1;Wert2;)

VERGLEICH(Text1;Text2)

WAHR()

WENN(Bedingung; Wert1; Wert2)

XOR(Wert1;Wert2;)

1.54 ÄHNLICH(Text1;Text2)

ÄHNLICH(Text1;Text2)

(Englisch: SIMILAR)

Überprüft, ob die beiden Texte Text1 und Text2 ähnlich sind und gibt entsprechend WAHR oder FALSCH zurück.

Beispiel:

ÄHNLICH("Beispiel";"Bsp") ergibt WAHR

Verwandte Funktionen:

ENTHÄLT, IDENTISCH, MUSTERVERGLEICH, VERGLEICH

1.55 ENTHÄLT(Text1;Text2)

ENTHÄLT(Text1;Text2)

(Englisch: CONTAINS)

Vergleicht die beiden Texte Text1 und Text2 und prüft, ob Text2 in Text1 enthalten ist.

Beispiele:

ENTHÄLT("Beispiel";"spiel") ergibt WAHR

ENTHÄLT("Beispiel";"spielt") ergibt FALSCH

Verwandte Funktionen:

ÄHNLICH, IDENTISCH, MUSTERVERGLEICH, VERGLEICH

1.56 FALSCH()

FALSCH()

(Englisch: FALSE)

Die Funktion FALSCH liefert den Wert FALSCH (=0) zurück. Sie kann entweder als FALSCH oder als FALSCH() geschrieben werden.

Beispiele:

WENN(FALSCH;1;2) ergibt 2

1.57 IDENTISCH(Text1;Text2)

IDENTISCH(Text1;Text2)

(Englisch: EXACT)

Vergleicht die beiden Texte Text1 und Text2. Es wird dabei überprüft, ob die beiden Texte identisch sind. Zwischen Groß- und Kleinschreibung wird also unterschieden, im Gegensatz zu VERGLEICH sowie dem direkten Vergleich mit "=".

Beispiele:

IDENTISCH("Beispiel";"Beispiel") ergibt WAHR

IDENTISCH("Beispiel";"BeiSpiel") ergibt FALSCH

Verwandte Funktionen:

ÄHNLICH, ENTHÄLT, MUSTERVERGLEICH, VERGLEICH

1.58 ISTDATUM(Wert)

ISTDATUM(Wert)

(Englisch: ISDATE)

Die Funktion ISTDATUM prüft, ob es sich bei Wert um ein Datum handelt. Ist dies der Fall, so wird WAHR zurückgegeben, ansonsten FALSCH.

Dies wird meist dazu benutzt, um zu prüfen, ob der Inhalt einer Zelle ein Datum ist.

Beispiele:

ISTDATUM(12) ergibt FALSCH

ISTDATUM(A1) ergibt WAHR, falls in A1 ein Datum steht.

1.59 ISTFEHLER(Bezug)

ISTFEHLER(Bezug)

(Englisch: ISERROR)

Die Funktion ISTFEHLER prüft, ob die Zelle Bezug eine Fehlermeldung enthält. Ist dies der Fall, so wird WAHR zurückgegeben, ansonsten FALSCH.

Beispiele:

ISTFEHLER(A1) mit A1 enthält "12" ergibt FALSCH

ISTFEHLER(A1) mit A1 enthält "=2+" ergibt WAHR

1.60 ISTGERADE(Zahl)

ISTGERADE(Zahl)

(Englisch: ISEVEN)

Diese Funktion liefert den Wert WAHR, wenn die als Argument angegebene Zahl gerade ist, oder FALSCH, wenn die Zahl ungerade ist.

Beispiele:

ISTGERADE(4) ergibt WAHR

ISTGERADE(17) ergibt FALSCH

Verwandte Funktionen:

ISTUNGERADE, ISTZAHL, GERADE, UNGERADE

1.61 ISTLEER(Bezug)

ISTLEER(Bezug)

(Englisch: ISEEMPTY)

Die Funktion ISTLEER prüft, ob die Zelle, auf die Bezug zeigt, leer ist. Ist dies der Fall, so wird WAHR zurückgegeben, ansonsten FALSCH.

Bezug: Ist ein Bezug auf eine Zelle, etwa A1

Beispiele:

ISTLEER(A1) ergibt FALSCH, falls dort etwa eine Zahl, ein Text oder etwas anderes steht.

1.62 ISTTEXT(Wert)

ISTTEXT(Wert)

(Englisch: ISSTRING)

Die Funktion ISTTEXT prüft, ob es sich bei Wert um einen Text handelt. Ist dies der Fall, so wird WAHR zurückgegeben, ansonsten FALSCH.

Beispiele:

ISTTEXT("12") ergibt WAHR

ISTTEXT(A1) ergibt WAHR, falls in A1 ein Text steht.

1.63 ISTUNGERADE(Zahl)

ISTUNGERADE(Zahl)

(Englisch: ISODD)

Diese Funktion liefert den Wert WAHR, wenn die als Argument angegebene Zahl ungerade ist, oder FALSCH, wenn die Zahl gerade ist.

Beispiele:

ISTUNGERADE(17) ergibt WAHR

ISTUNGERADE(4) ergibt FALSCH

Verwandte Funktionen:

ISTGERADE, ISTZAHL, GERADE, UNGERADE

1.64 ISTZAHL(Wert)

ISTZAHL(Wert)

(Englisch: ISVALUE)

Die Funktion ISTZAHL prüft, ob es sich bei Wert um eine Zahl handelt. Ist dies der Fall, so wird WAHR zurückgegeben, ansonsten FALSCH.

Beispiele:

ISTZAHL("12") ergibt FALSCH

ISTZAHL(A1) ergibt WAHR, falls in A1 eine Zahl steht.

1.65 ISTZEIT(Wert)

ISTZEIT(Wert)

(Englisch: ISTIME)

Die Funktion ISTZEIT prüft, ob es sich bei Wert um eine Zeit handelt. Ist dies der Fall, so wird WAHR zurückgegeben, ansonsten FALSCH.

Beispiele:

ISTZEIT(12) ergibt FALSCH

ISTZEIT(A1) ergibt WAHR, falls in A1 eine Zeit steht.

1.66 MUSTERVERGLEICH(Text;Muster)

MUSTERVERGLEICH(Text;Muster)

(Englisch: MATCHES)

Überprüft, ob Text dem Muster entspricht. Zwischen Groß- und Kleinschreibung wird nicht unterschieden.

Das Muster kann dabei folgende Vergleichskriterien aufweisen:

* entspricht bel. vielen (mind. einem) beliebigen Zeichen

? entspricht genau einem beliebigen Zeichen

Beispiele:

MUSTERVERGLEICH("Beispiel";"*spiel") ergibt WAHR

MUSTERVERGLEICH("Beispiel";"B??SPIEL") ergibt WAHR

Verwandte Funktionen:

ÄHNLICH, ENTHÄLT, IDENTISCH, VERGLEICH

1.67 NICHT(Wert)

NICHT(Wert)

(Englisch: NOT)

Die Funktion NICHT negiert den Wahrheitswert, d.h. ist Wert WAHR, dann wird FALSCH zurückgegeben und umgekehrt.

Wert: Kann ein Wahrheitswert oder eine Zahl sein. 0 stellt dann FALSCH dar, alle anderen Zahlen ergeben WAHR.

Beispiele:

NICHT(WAHR) ergibt FALSCH

NICHT(FALSCH) ergibt WAHR

NICHT(3) ergibt FALSCH

NICHT(WAHR UND FALSCH) ergibt WAHR

Verwandte Funktionen:

UND, ODER, XOR

1.68 ODER(Wert1;Wert2;)

ODER(Wert1;Wert2;)

(Englisch: OR)

Die Funktion ODER gibt die Oder-Verknüpfung aller Werte als Wahrheitswert zurück, d.h. es wird WAHR zurückgegeben, falls mindestens einer der Werte WAHR oder nicht Null ist. Sind alle Werte Null oder FALSCH, so wird FALSCH zurückgegeben.

Werte: Dies können beliebig viele, durch Semikolon getrennte Werte sein, und zwar: Wahrheitswerte oder Zahlen. (eine Zahl gilt dann als Wahr, falls sie nicht 0 ist)

Hinweis: Statt dieser Funktionsschreibweise kann ODER (oder OR) auch einfach als Operator (wie etwa "+" benutzt werden), also: A1 ODER A3. Dies ist vor allem dann sinnvoller und übersichtlicher, wenn nur zwei Werte oder-verknüpft werden sollen. (Weiterhin kann in dieser Schreibweise OR auch als Oder-Verknüpfung zweier Zahlen betrachtet werden: 3 OR 4 ergibt dann 7)

Hinweis: Ab TurboCalc4 können für die Funktionen UND, ODER und XOR auch Bereiche als Parameter angegeben werden (wie etwa bei SUMME), siehe letztes Beispiel.

Beispiele:

ODER(WAHR;FALSCH) ergibt WAHR

ODER(FALSCH;0;0) ergibt FALSCH

ODER(FALSCH;0;1) ergibt WAHR

WENN(ODER(0;1);3;4) ergibt 4.

dies entspricht auch:

WAHR ODER FALSCH,

FALSCH ODER 0 ODER 0

FALSCH ODER 0 ODER 1

WENN(0 ODER 3;3;4)

2 OR 1 ergibt 3 und 3 OR 2 ergibt 3

ODER(A1:C5) Prüft, ob mindestens eine nichtleere Zellen WAHR bzw. eine Zahl ungleich Null enthält.

Verwandte Funktionen:

UND, NICHT, XOR

1.69 UND(Wert1;Wert2;)

UND(Wert1;Wert2;)

(Englisch: AND)

Die Funktion UND gibt die Und-Verknüpfung aller Werte als Wahrheitswert zurück, d.h. es wird WAHR zurückgegeben, falls alle Werte WAHR oder nicht Null sind. Ist auch nur ein Wert FALSCH oder Null, so wird FALSCH zurückgegeben.

Werte: Dies können beliebig viele, durch Semikolon getrennte Werte sein, und zwar: Wahrheitswerte oder Zahlen. (eine Zahl gilt dann als Wahr, falls sie nicht 0 ist)

Hinweis: Statt dieser Funktionsschreibweise kann UND (oder AND) auch einfach als Operator (wie etwa "+" benutzt werden), also: A1 UND A3. Dies ist vor allem dann sinnvoller und übersichtlicher, wenn nur zwei Werte und-verknüpft werden sollen. (Weiterhin kann in dieser Schreibweise UND auch als Und-Verknüpfung zweier Zahlen betrachtet werden: 7 UND 4 ergibt dann 4)

Hinweis: Ab TurboCalc4 können für die Funktionen UND, ODER und XOR auch Bereiche als Parameter angegeben werden (wie etwa bei SUMME), siehe letztes Beispiel.

Beispiele:

UND(WAHR;FALSCH) ergibt FALSCH

UND(WAHR;1;2) ergibt WAHR

UND(FALSCH;WAHR;1) ergibt FALSCH

WENN(UND(0;1);3;4) ergibt 4.

dies entspricht auch:

WAHR UND FALSCH,

WAHR UND 1 UND 2

FALSCH UND WAHR UND 1

WENN(0 UND 1;3;4)

3 UND 1 ergibt 1

7 UND 3 ergibt 3

UND(A1:C5) Prüft, ob alle nichtleeren Zellen WAHR bzw. eine Zahl ungleich Null enthalten.

Verwandte Funktionen:

ODER, NICHT, XOR

1.70 VERGLEICH(Text1;Text2)

VERGLEICH(Text1;Text2)

(Englisch: COMPARE)

Vergleicht die beiden Texte Text1 und Text2. Zwischen Groß- und Kleinschreibung wird dabei im Gegensatz zu IDENTISCH nicht unterschieden; diese Funktion entspricht dem direkten Vergleich mit "=".

Beispiele:

VERGLEICH("Beispiel";"Beispiel") ergibt WAHR

VERGLEICH("Beispiel";"BeiSpiel") ergibt WAHR

Verwandte Funktionen:

ÄHNLICH, ENTHÄLT, IDENTISCH, MUSTERVERGLEICH

1.71 WAHR()

WAHR()

(Englisch: TRUE)

Die Funktion WAHR liefert den Wert WAHR (=1) zurück. Sie kann entweder als WAHR oder als WAHR() geschrieben werden.

Beispiele:

WENN(WAHR;1;2) ergibt 1

1.72 WENN(Bedingung; Wert1; Wert2)

WENN(Bedingung; Wert1; Wert2)

(Englisch: IF)

Die Funktion WENN gibt in Abhängigkeit von Bedingung entweder Wert1 (falls Bedingung erfüllt, WAHR) oder aber Wert2 zurück.

Bedingung kann dabei ein beliebiger Bool'scher Ausdruck sein, d.h. Vergleich ggf. mit UND oder ODER verknüpft.

Wert1, Wert2 können beliebige Ausdrücke (Texte, Zahlen...) sein. (Auch weitere WENN(...;...;...) als Parameter sind möglich - WENN läßt sich also schachteln).

Tip: Diese Funktion ist sehr praktisch, da damit auf einfache Weise Fallunterscheidungen gemacht werden können.

Beispiele:

WENN(WAHR;"Hallo";"Du") ergibt "Hallo"

WENN(1=3;"Hallo";"Du") ergibt "Du"

WENN(A1>0;"Gewinn";"Verlust") bestimmt in Abhängigkeit von A1, ob Gewinn oder Verlust gemacht wurde.

WENN(A1<10;25;25+(A1-10)*0.12) berechnet die Telefonkosten bei A1 Einheiten (mit Grundgebühr: 25 DM, Einheit 12 Pf und 10 Freieinheiten)

Verwandte Funktionen:

AUSWAHL

1.73 XOR(Wert1;Wert2;)

XOR(Wert1;Wert2;)

Die Funktion XOR gibt die Exklusiv-Oder-Verknüpfung aller Werte als Wahrheitswert zurück, d.h. es wird FALSCH zurückgegeben, falls eine gerade Anzahl von Werten WAHR oder nicht Null ist. Sonst wird WAHR zurückgegeben.

Wert1 Wert2 Wert1 XOR Wert2

FALSCH FALSCH FALSCH

FALSCH WAHR WAHR

WAHR FALSCH WAHR

WAHR WAHR FALSCH

Werte: Dies können beliebig viele, durch Semikolon getrennte Werte sein, und zwar: Wahrheitswerte oder Zahlen. (eine Zahl gilt dann als WAHR, falls sie nicht 0 ist)

Hinweis: Statt dieser Funktionsschreibweise kann XOR auch einfach als Operator (wie etwa "+") benutzt werden, also: A1 XOR A3. Dies ist vor allem dann sinnvoller und übersichtlicher, wenn nur zwei Werte exklusiv-oder-verknüpft werden sollen. (Weiterhin kann in dieser Schreibweise XOR auch als Exklusiv-Oder-Verknüpfung zweier Zahlen betrachtet werden: 3 XOR 4 ergibt dann 7.)

Hinweis: Ab TurboCalc4 können für die Funktionen UND, ODER und XOR auch Bereiche als Parameter angegeben werden (wie etwa bei SUMME), siehe letztes Beispiel.

Beispiele:

XOR(WAHR;FALSCH) ergibt WAHR

XOR(FALSCH;0;0) ergibt FALSCH

XOR(FALSCH;0;1) ergibt WAHR

WENN(XOR(0;1);3;4) ergibt 3.

dies entspricht auch:

WAHR XOR FALSCH,

FALSCH XOR 0 XOR 0

FALSCH XOR 0 XOR 1

WENN(0 XOR 3;3;4)

2 XOR 1 ergibt 3

3 XOR 2 ergibt 1

XOR(A1:C5) Prüft, ob die Anzahl der nichtleeren Zellen, die WAHR bzw. eine Zahl ungleich Null enthalten, gerade ist.

Verwandte Funktionen:

ODER; UND, NICHT

1.74 Text-Funktionen

Text-Funktionen

Plus: +

BEHALTEN(String;Liste)

BERECHNEN(Formel)

CODE(Text)

BIN(Zahltext)
CONV2INT(Zahltext;Basis)
CONV2TEXT(Zahl;Basis)
ERSETZEN(String;Text1[;Text2])
GLÄTTEN(Text)
GROSS(Text)
GROSS2(Text)
HEX(Zahltext)
INTEXT(String;Muster[;Pos])
KLEIN(Text)
KOMPRIMIEREN(String[;Liste])
LÄNGE(Text)
LINKS(Text;Anzahl)
MITTE(Text;Zahl1;Zahl2)
NACHBIN(Zahl)
NACHHEX(Zahl)
NACHOCT(Zahl)
OCT(Zahltext)
RECHTS(Text;Anzahl)
RÖMISCH(Zahl)
SÄUBERN(Text)
SCHIEBENL(Text)
SCHIEBENR(Text)
SPIEGELN(Text)
TEIL(Text;Zahl1;Zahl2)
TEXT(Data[;Format])
WERT(Text)
WIEDERHOLEN(Text; Anzahl)
WORT(Text;Num)
WÖRTER(Text)
ZEICHEN(asc-code)

1.75 Plus: +

Plus: +

Der Operator "+" kann zum Verketteten (Hintereinanderschreiben) von Texten benutzt werden. Er wird dabei genauso wie bei normalen Zahlen benutzt:

Hinweis: Eine Minusoperation sowie eine Operation zwischen einem Text und einer Zahl ist dabei nicht definiert, es erscheint dann #TYP.

Beispiele:

"Hallo" + " " + "Welt" ergibt "Hallo Welt"

"Ab"+"C" ergibt "AbC"

1.76 BEHALTEN(String;Liste)

BEHALTEN(String;Liste)

(Englisch: KEEP)

Entfernt alle Vorkommen der Zeichen String die nicht in Liste enthalten sind und gibt das Ergebnis zurück. (Wird Liste ausgelassen, so werden alle String nicht verändert).

String: Text, der modifiziert werden soll.

Liste: Text mit allen Zeichen, die nicht entfernt werden sollen.

Beispiele:

BEHALTEN("Dies ist ein Beispiel";"Beis") ergibt "iesiseiBeisie"

Verwandte Funktionen:

ERSETZEN, **KOMPRIMIEREN**

1.77 BERECHNEN(Formel)

BERECHNEN(Formel)

(Englisch: EVAL)

Diese Funktion berechnet Formel. Im Text Formel können dabei alle TurboCalc-Funktionen und -Operatoren verwendet werden.

Formel: Der Text der ausgewertet und berechnet werden soll.

Beispiele:

BERECHNEN("2+2") ergibt 4

1.78 CODE(Text)

CODE(Text)

Die Funktion CODE liefert den ASCII-Code (American Standard Code for Information Interchange - siehe Handbuch zum Amiga) des ersten Zeichens des Texts.

Beispiele:

CODE("A") ergibt 65

CODE("ABC") ergibt 65

CODE("a") ergibt 97 (da klein geschrieben!)

Verwandte Funktionen:

ZEICHEN

1.79 BIN(Zahltext)

BIN(Zahltext)

Diese Funktion wandelt den Text Zahltext aus dem Binärsystem in eine 'normale' Zahl im Dezimalsystem um.

Zahltext ist ein Text der aus den Ziffern '0'-'1' bestehen sollte. Jedes andere Zeichen beendet die Umwandlung.

Beispiele:

BIN("1001") ergibt 9

Verwandte Funktionen:

OCT, **HEX**

1.80 CONV2INT(Zahltext;Basis)

CONV2INT(Zahltext;Basis)

Diese Funktion wandelt den Text Zahltext aus dem Zahlensystem mit der angegebene Basis in eine 'normale' Zahl im Dezimalsystem um.

Zahltext ist ein Text der aus den Ziffern '0'-'9' sowie den Zeichen 'A'-'Z' (je nach Basis) Jedes andere Zeichen beendet die Umwandlung.

Beispiele:

CONV2INT("123";8) ergibt 83

CONV2INT(TEXT(123);7) ergibt 66

Verwandte Funktionen:

BIN, HEX, OCT, CONV2TEXT

1.81 CONV2TEXT(Zahl;Basis)

CONV2TEXT(Zahl;Basis)

Diese Funktion wandelt Zahl aus dem Dezimalsystem in einen Text um, der die Zahl im mit Basis angegebenen Zahlensystem darstellt.

Zahl ist eine beliebige positive ganze Zahl.

Beispiele:

CONV2TEXT(83;8) ergibt 123

CONV2TEXT(66;7) ergibt 123

Verwandte Funktionen:

BIN, HEX, OCT, CONV2INT

1.82 ERSETZEN(String;Text1[:Text2])

ERSETZEN(String;Text1[:Text2])

(Englisch: TRANSLATE)

Ersetzt in String alle Vorkommen der Zeichen aus Text1 durch das entsprechende Zeichen von Text2. Die anderen Zeichen von String werden nicht verändert.

String: Text, der modifiziert werden soll.

Text1: Text mit allen Zeichen, die durch Zeichen aus Text2 ersetzt werden sollen.

Text2: Text mit den Zeichen, durch die das entsprechende Zeichen aus Text1 ersetzt werden.

Enthält Text1 weniger Zeichen als Text2, so werden die überzähligen Zeichen von Text2 ignoriert.

Enthält Text1 mehr Zeichen als Text2, so werden die Vorkommen der Zeichen, für die es kein passendes Zeichen in Text2 gibt, gelöscht.

Wird Text2 ausgelassen, so entspricht diese Funktion KOMPRIMIEREN

Beispiele:

ERSETZEN("Dies ist ein Text";"s";"x") ergibt "Diex ixt ein Text"

ERSETZEN("Dies ist ein Text";"sie";"x") ergibt "Dx xt n Txt"

Verwandte Funktionen:

BEHALTEN, KOMPRIMIEREN

1.83 GLÄTTEN(Text)

GLÄTTEN(Text)

(Englisch: TRIM)

Die Funktion GLÄTTEN löscht aus Text alle "unnötigen" Leerzeichen, d.h. es läßt nur noch jeweils ein Leerzeichen übrig.

Beispiele:

GLÄTTEN("Januar Februar März") ergibt "Januar Februar März"

Verwandte Funktionen:

SÄUBERN

1.84 GROSS(Text)

GROSS(Text)

(Englisch: UPPER)

Die Funktion GROSS wandelt alle Buchstaben des Texts in Großbuchstaben um.

Beispiele:

GROSS("Dies ist ein Text") liefert "DIES IST EIN TEXT"

Verwandte Funktionen:

GROSS2, KLEIN

1.85 GROSS2(Text)

GROSS2(Text)

(Englisch: UPPER2)

Die Funktion GROSS2 wandelt die ersten Buchstaben jedes Wortes in Groß- und alle anderen in Kleinbuchstaben um.

Beispiele:

GROSS2("Dies ist ein Text") liefert "Dies Ist Ein Text"

Verwandte Funktionen:

GROSS, KLEIN

1.86 HEX(Zahltext)

HEX(Zahltext)

Diese Funktion wandelt den Text 'Zahltext' aus dem Hexadezimalsystem in eine 'normale' Zahl im Dezimalsystem um.

Zahltext ist ein Text der aus den Ziffern '0'-'9' und den Zeichen 'A'-'F' bestehen sollte (Kleinschreibung natürlich möglich). Jedes andere Zeichen beendet die Umwandlung.

Beispiele:

HEX("C8") ergibt 200

HEX("C8X1") ergibt 200, die Umwandlung wird beim 'X' abgebrochen.

Verwandte Funktionen:

BIN, OCT

1.87 INTEXT(String;Muster[;Pos])

INTEXT(String;Muster[;Pos])

(Englisch: INSTRING)

Gibt die erste Position (ab der Stelle Pos) zurück, ab der Muster in String zu finden ist. Konnte der Text nicht gefunden werden, so wird 0 zurückgegeben.

String: zu durchsuchender Text.

Muster: Text, der in String gesucht werden soll.

Pos: Die Position, ab der die Suche begonnen werden soll (1=erste Position - wird dieser Parameter ausgelassen, so wird ab dem Beginn durchsucht)

Beispiele:

INTEXT("ein Beispieltext","ei") ergibt 1

INTEXT("ein Beispieltext","ei";2) ergibt 6

INSTRING("ein Beispieltext","abc") ergibt 0 (d.h. nicht gefunden)

1.88 KLEIN(Text)

KLEIN(Text)

(Englisch: LOWER)

Die Funktion KLEIN wandelt alle Buchstaben des Textes in Kleinbuchstaben um.

Beispiele:

KLEIN("Dies ist ein Text") liefert "dies ist ein text"

Verwandte Funktionen:

GROSS, GROSS2

1.89 KOMPRIMIEREN(String[;Liste])

KOMPRIMIEREN(String[;Liste])

(Englisch: COMPRESS)

Entfernt alle Vorkommen der Zeichen aus Liste aus String und gibt diesen zurück. (Wird Liste ausgelassen, so werden alle Leerzeichen entfernt).

String: Text, der komprimiert werden soll.

Liste: Text mit allen Zeichen, die entfernt werden sollen. Wird dieser Parameter ausgelassen, so wird " " (Leerzeichen) angenommen.

Beispiele:

KOMPRIMIEREN(" a b c") ergibt "abc"

KOMPRIMIEREN("++a-b++c";"+-") ergibt "abc"

Verwandte Funktionen:

ERSETZEN, BEHALTEN

1.90 LÄNGE(Text)

LÄNGE(Text)

(Englisch: LENGTH)

Die Funktion LÄNGE liefert die Anzahl der Zeichen des Textes als Zahl.

Beispiele:

LÄNGE("Michael Friedrich") ergibt 17

LÄNGE("") liefert 0

1.91 LINKS(Text;Anzahl)

LINKS(Text;Anzahl)

(Englisch: LEFT)

Die Funktion LINKS liefert die ersten Anzahl Zeichen des Textes (von links)

Anzahl muß dabei größer als 0 sein (sonst erscheint die Fehlermeldung #WERT). Ist Anzahl größer als die Länge des Textes, so wird der Text zurückgegeben.

Wird Anzahl ausgelassen (also LINKS(Text)), so wird das erste Zeichen des Textes zurückgegeben.

Beispiele:

LINKS("Michael Friedrich";4) ergibt "Mich"

LINKS("Hallo") ergibt "H"

Verwandte Funktionen:

MITTE, TEIL, RECHTS

1.92 MITTE(Text;Zahl1;Zahl2)

MITTE(Text;Zahl1;Zahl2)

(Englisch: MID)

Die Funktion MITTE liefert ab der Stelle Zahl1 einen höchstens Zahl2-langen Teiltext von Text.

Zahl1 muß größer als 0 sein (ansonsten wird der Fehler #WERT zurückgegeben) und bestimmt die Position, ab der der Text entnommen werden soll.

Zahl2 muß größer 0 sein (auch sonst: #WERT) und bestimmt, wie viele Zeichen entnommen werden sollen. Ist Zahl2 größer als die restliche Textlänge, so wird dieser Resttext zurückgegeben.

Beispiele:

MITTE("Michael Friedrich";2;3) liefert "ich"

MITTE("Michael Friedrich";8;20) liefert "Friedrich"

MITTE("Michael Friedrich";20;20) liefert ""

Verwandte Funktionen:

LINKS, RECHTS, TEIL

1.93 NACHBIN(Zahl)

NACHBIN(Zahl)

(Englisch: TOBIN)

Diese Funktion wandelt Zahl in das Binärsystem um und liefert einen Text zurück.

Beispiele:

NACHBIN(17) ergibt 10001

Verwandte Funktionen:

BIN, **NACHOCT**, **NACHHEX**

1.94 NACHHEX(Zahl)

NACHHEX(Zahl)

(Englisch: TOHEX)

Diese Funktion wandelt Zahl in das Hexadezimalsystem um und liefert als Ergebnis den entsprechenden Text zurück.

Beispiele:

NACHHEX(15) ergibt f

Verwandte Funktionen:

HEX, **NACHBIN**, **NACHOCT**

1.95 NACHOCT(Zahl)

NACHOCT(Zahl)

(Englisch: TOOCT)

Diese Funktion wandelt Zahl in das Oktalsystem um und liefert als Ergebnis den entsprechenden Text zurück.

Beispiele:

NACHOCT(15) ergibt 17

Verwandte Funktionen:

OCT, **NACHBIN**, **NACHHEX**

1.96 OCT(Zahltext)

OCT(Zahltext)

Diese Funktion wandelt den Text Zahltext aus dem Oktalsystem in eine 'normale' Zahl im Dezimalsystem um.

Zahltext ist ein Text der aus den Ziffern '0'-'7' bestehen sollte. Jedes andere Zeichen beendet die Umwandlung.

Beispiele:

OCT("123") ergibt 83

OCT(TEXT(123)) ergibt 83

Verwandte Funktionen:

BIN, **HEX**

1.97 RECHTS(Text;Anzahl)

RECHTS(Text;Anzahl)

(Englisch: RIGHT)

Die Funktion RECHTS liefert die letzten Anzahl Zeichen des Textes (von rechts)

Anzahl muß dabei größer als 0 sein (sonst erscheint die Fehlermeldung #WERT). Ist Anzahl größer als der Länge des Textes, so wird der Text zurückgegeben.

Wird Anzahl ausgelassen (also RECHTS(Text)), so wird das letzte Zeichen des Textes zurückgegeben.

Beispiele:

RECHTS("Michael Friedrich";3) ergibt "ich"

RECHTS("Hallo") ergibt "o"

Verwandte Funktionen:

LINKS, MITTE, TEIL

1.98 RÖMISCH(Zahl)

RÖMISCH(Zahl)

(Englisch: ROMAN)

Wandelt Zahl in eine römische Zahl als Text um.

Hat Zahl den Wert 0, so wird ein leerer Text ("") zurückgegeben; ist Zahl negativ oder zu groß (siehe folgender Hinweis), so wird der Fehler #WERT retourniert.

Hinweis: Würden zur Darstellung der Zahl mehr als 90 Textstellen benötigt, so wird der Fehlerwert #WERT zurückgegeben.

Beispiele:

RÖMISCH(1996) ergibt MCMXCVI

RÖMISCH(11) ergibt XI

1.99 SÄUBERN(Text)

SÄUBERN(Text)

(Englisch: CLEAN)

Die Funktion SÄUBERN entfernt aus dem Text alle nicht druckbaren Zeichen, also Steuercodes. Dies ist sinnvoll, wenn Sie einen Text importiert haben und dieser Steuerzeichen enthielt (dies ist am Bildschirm oft an einem rechteckigen Kasten zu erkennen).

Beispiele:

SÄUBERN(ZEICHEN(7)+"Text") liefert "Text"

Verwandte Funktionen:

ZEICHEN, GLÄTTEN

1.100 SCHIEBENL(Text)

SCHIEBENL(Text)

(Englisch: SHIFTL)

Verschiebt die Buchstabenreihenfolge um eins nach links (d.h. das zweite Zeichen wird erstes), das Zeichen ganz links wird am Ende angefügt.

Text: ein beliebiger Text.

Beispiele:

SCHIEBENL("Hallo") ergibt "alloH"

1.101 SCHIEBENR(Text)

SCHIEBENR(Text)

(Englisch: SHIFTR)

Verschiebt die Buchstabenreihenfolge um eins nach rechts, d.h. das letzte Zeichen wird entfernt und am Anfang angefügt.

Text: ein beliebiger Text.

Beispiele:

SCHIEBENR("Hallo") ergibt "oHall"

1.102 SPIEGELN(Text)

SPIEGELN(Text)

(Englisch: REVERSE)

Dreht die Buchstabenreihenfolge von Text um.

Text: ein beliebiger Text.

Beispiele:

SPIEGELN("Hallo") ergibt "ollaH"

1.103 TEIL(Text;Zahl1;Zahl2)

TEIL(Text;Zahl1;Zahl2)

(Englisch: PART)

Diese Funktion ist äquivalent zu MITTE. Sie wurde nur zur Kompatibilität aufgenommen, Beschreibung siehe bei MITTE.

1.104 TEXT(Data[;Format])

TEXT(Data[;Format])

Die Funktion TEXT wandelt alle Datentypen (Zahlen, Wahrheitswerte, Datum, Uhrzeit und Texte) in Text um.

Data: Kann ein beliebiger Ausdruck sein.

Format: Gibt das Format an, wie Data als Text formatiert werden soll. 0 bzw. Auslassen des Parameters entspricht dem Standardformat. Die weiteren Formatnummern können Sie in der Tabelle bei Makrobefehl ZAHLENFORMAT ablesen. (bzw. im Fenster zu <Format-Zahlenformat> abzählen und dabei bei 0 beginnen).

Diese Funktion ist somit das Gegenstück zu WERT.

Beispiele:

TEXT(123;3) ergibt "123.00"

TEXT(12)+ " DM" ergibt "12 DM"

WERT(TEXT(123.45;1)) ergibt 123 (da TEXT(123.45;1) "123" erzeugt).

1.105 WERT(Text)

WERT(Text)

(Englisch: VALUE)

Die Funktion Wert wandelt den Text in eine Zahl, ein Datum, eine Uhrzeit oder einen Fehlerwert um.

Dies ist sinnvoll, wenn man Datums- oder Zeitwerte in Formeln eingeben möchte, da dort eine normale Eingabe nicht möglich ist. Weiterhin kann diese Funktion gut benutzt werden, bestimmte Zellinhalte zu prüfen, ohne eine Fehlermeldung zu riskieren (enthalten sie Zahlen, so werden diese einfach weitergegeben, bei normalen Texten wird 0 zurückgegeben).

Diese Funktion ist somit das Gegenstück zu TEXT.

Wert: kann als Parameter auch eine Zahl, Boolean, Datum, Uhrzeit sein. In diesem Fall wird dann dies entsprechend zurückgegeben. (WERT(42)=42, WERT(B1) mit B1=13:02 ergibt 13:02)

Damit können jetzt sehr leicht Zellinhalte, die sowohl Zahlen als auch Texte enthalten können, getestet werden, ohne einen #TYP Fehler zu erhalten. (etwa: SPRINGEWENN(WERT(b1)=0;c5) - springt wenn B1 0 ist oder Text enthält, der nicht als Zahl interpretiert werden kann)

Beispiele:

WERT("1000") ergibt 1000.

WERT("20 Jan") liefert 20.01.1993 (im Jahr 1993)

WERT(123) ergibt 123

Enthält die Zelle A1 ein Datum, so kann man damit die Tage seit 1.1.1992 berechnen; A1-WERT("1.1.1992")

1.106 WIEDERHOLEN(Text; Anzahl)

WIEDERHOLEN(Text; Anzahl)

(Englisch: REPEAT)

Die Funktion WIEDERHOLEN liefert als Ergebnis einen Text, der aus Anzahl Wiederholungen des Parameters Text besteht.

Anzahl muß dabei größer als 0 sein, ansonsten wird #WERT zurückgegeben.

Beispiele:

WIEDERHOLEN("-";5) ergibt "--"

WIEDERHOLEN("Text";3) ergibt "TextTextText"

1.107 WORT(Text;Num)

WORT(Text;Num)

(Englisch: WORD)

Diese Funktion trennt ein Wort aus einem Text heraus und gibt es zurück.

Als Trennzeichen zwischen Wörtern gilt dabei das Leerzeichen, Tab (ASCII-Wert 9) sowie die Interpunktionszeichen ".,-!?".

Ein Wert von Num außerhalb des gültigen Bereichs (1 bedeutet erstes Wort) führt zur Fehlermeldung #WERT.

Beispiele:

WORT("Dies ist ein Test.";3) ergibt "ein"

Verwandte Funktionen:

WÖRTER, LÄNGE

1.108 WÖRTER(Text)

WÖRTER(Text)

(Englisch: WORDS)

Diese Funktion ermittelt die Anzahl der Wörter in Text.

Als Trennzeichen zwischen Wörtern gilt dabei das Leerzeichen, Tab (ASCII-Wert 9) sowie die Interpunktionszeichen ".,-!?".

Beispiele:

WÖRTER("Dies ist ein Test.") ergibt 4

Verwandte Funktionen:

WORT, LÄNGE

1.109 ZEICHEN(asc-code)

ZEICHEN(asc-code)

(Englisch: CHAR)

Die Funktion ZEICHEN liefert das Zeichen (d.h. Text der Länge 1), den die Zahl asc-code repräsentiert. (Hierzu wird der ASCII-Code benutzt). Dies ist also die Umkehrung zu CODE.

asc-code muß sich zwischen 1 und 255 befinden!

Beispiele:

ZEICHEN(65) ergibt "A"

ZEICHEN(CODE("T")) ergibt "T"

Verwandte Funktionen:

CODE

1.110 Datum- und Zeit-Funktionen

Datum- und Zeit-Funktionen

Den meisten der unten aufgeführten Funktionen muß ein Datum bzw. eine Uhrzeit übergeben werden (oder sie liefern diese zurück).

Plus: +, Minus: -

DATUM(Jahr;Monat;Tag)

DATWERT(Text)

HEUTE()

JAHR(Datum)

JETZT()

MINUTE(Zeit)

MINUTEN(Zeit)

MONAT(Datum)

MONATE(Datum)

MONATTEXT(Datum[:Abkürzung])

SEKUNDE(Zeit)

SEKUNDEN(Zeit)

STUNDE(Zeit)

STUNDEN(Zeit)

TAG(Datum)

TAGTEXT(Datum[:Abkürzung])

WOCHENTAG(Datum)

ZEIT(Stunde;Minute;Sekunde)

ZEITWERT(Text)

Deswegen hier zu diesen beiden Werten einige Informationen:

Datum: Das Datum wird intern als laufende Ganzzahl gespeichert, welches der Zahl 0 den 1.1.1900 zuweist. So hat etwa das Datum 1.7.93 den Wert 34149.

(Mittels <Format-Zahlenformat> kann man die Anzeige als Zahl ("0") oder als Datum ("TT-MM-JJJJ") erzwingen - Standard entscheidet selbständig).

Diese Möglichkeit kann man für fortschrittliche Anwendungen benutzen (etwa ergibt "Datum-Datum" gerade die Anzahl der Tage zwischen den beiden Tagen, siehe unten.)

Uhrzeit: Entsprechend wird auch die Uhrzeit gespeichert: Als Anzahl der Sekunden seit 0 Uhr!

Somit kann meist, falls ein Datum als Parameter verlangt wird, auch eine normale Zahl eingegeben werden. Beachten Sie jedoch dann die Interpretation dieser Zahl!

1.111 Plus: +, Minus: -

Plus: +, Minus: -

Die Operationen "+" und "-" sind auch für das Datum und die Uhrzeit definiert, und zwar nach folgenden Regeln:

Datum + Zahl = Datum (Datum um Zahl Tage erhöhen)

Zahl + Datum = Datum (das gleiche)

Datum - Zahl = Datum (Datum um Zahl Tage vermindern)

Datum - Datum = Zahl (Anzahl der Tage zwischen den beiden Daten)

(dabei stellt bei Zahl der Wert 1 genau einen Tag dar - Kommawerte werden ignoriert)

Uhrzeit + Zahl = Uhrzeit (Uhrzeit um Zahl Sekunden erhöhen)

Zahl + Uhrzeit = Uhrzeit (das gleiche)

Uhrzeit - Zahl = Uhrzeit (Uhrzeit um Zahl Sekunden vermindern)

Uhrzeit - Uhrzeit = Zahl (Anzahl der Sekunden zwischen den beiden Uhrzeiten)

(dabei stellt bei Zahl der Wert 1 genau eine Sekunde dar - Kommawerte werden ignoriert)

Hinweis: Bei anderen Operationen (etwa ZAHL-DATUM) wird das Datum in eine Zahl gewandelt und dann normal mit der Zahl weitergerechnet.

Beispiele:

WERT("5-1-1993")-WERT("1-1-1993") liefert 4

WERT("1-1-1993"+4) ergibt 5-4-1993

WERT("18:00")-WERT("12:00") ergibt 21600 (=6*60*60)

Verwandte Funktionen:

WERT, DATWERT, ZEITWERT, DATUM

1.112 DATUM(Jahr;Monat;Tag)

DATUM(Jahr;Monat;Tag)

(Englisch: DATE)

Die Funktion DATUM wandelt die drei Parameter in ein Datum um.

Jahr, Monat und Tag müssen dabei einen existierenden Tag ab 1.1.1900 bis 31.12.2400 darstellen - ansonsten wird der Fehler #WERT zurückgegeben.

Beispiele:

DATUM(1993;4;1) ergibt den 1. April 1993

Verwandte Funktionen:

DATWERT, TAG, MONAT, JAHR, HEUTE, JETZT, ZEITWERT

1.113 DATWERT(Text)

DATWERT(Text)

(Englisch: DATEVALUE)

Die Funktion DATWERT wandelt einen Text in ein Datum um. (Es kann auch einfach WERT benutzt werden, was alle Formate umwandeln kann, siehe dort.)

Text kann ein beliebiges Datumsformat in Textform sein, etwa:

"30-9-93"

"30 Sep 93"

"Sep 93"

"30 Sep"

(eine vollständige Beschreibung aller Möglichkeiten siehe im Kapitel "Die Eingabe")

Tip: Diese Funktion ist sehr praktisch, wenn man Datumswerte in Formeln verwenden möchte, da man obige Daten nicht einfach direkt eingeben kann.

Beispiele:

DATWERT("1.4.93") ergibt den 1. April 93

Verwandte Funktionen:

WERT, DATUM, HEUTE, ZEITWERT

1.114 HEUTE()

HEUTE()

(Englisch: TODAY)

Die Funktion HEUTE liefert als Ergebnis das Datum des heutigen Tages.

Hinweis: Achten Sie darauf, daß das Datum auch stimmt, falls Sie diese Funktion benutzen. Haben Sie eine interne (gepufferte) Uhr eingebaut, so sollte dies der Fall sein. Ansonsten sollten Sie das Datum mittels DATE per CLI oder aber per Preferences einstellen, wenn Sie sinnvolle Ergebnisse wünschen.

Beispiele:

HEUTE()+2 liefert das Datum von Übermorgen.

Verwandte Funktionen:

DATUM, DATWERT, JETZT

1.115 JAHR(Datum)

JAHR(Datum)

(Englisch: YEAR)

Die Funktion JAHR liefert von einem Datum die Jahreszahl als Zahl zurück.

Beispiele:

JAHR(WERT("23-3-93")) ergibt 1993

JAHR(2000) = 1905

Verwandte Funktionen:

MONAT, TAG, WOCHENTAG, HEUTE, JETZT

1.116 JETZT()

JETZT()

(Englisch: NOW)

Die Funktion JETZT liefert als Ergebnis die aktuelle Uhrzeit zurück.

Hinweis: Achten Sie darauf, daß die Uhrzeit auch stimmt, falls Sie diese Funktion benutzen. Haben Sie eine interne (gepufferte) Uhr eingebaut, so sollte dies der Fall sein. Ansonsten sollten Sie das Datum und die Uhrzeit mittels DATE per CLI oder aber per Preferences einstellen, wenn Sie sinnvolle Ergebnisse wünschen.

Beispiele:

JETZT() liefert die aktuelle Uhrzeit.

Verwandte Funktionen:

ZEITWERT, HEUTE

1.117 MINUTE(Zeit)

MINUTE(Zeit)

Diese Funktion extrahiert aus Zeit den Minutenwert. Der Stundenwert wird dabei ignoriert.

Beispiele:

MINUTE("15:23:02") ergibt 23

Verwandte Funktionen:

SEKUNDE, SEKUNDEN, MINUTEN, STUNDE, STUNDEN

1.118 MINUTEN(Zeit)

MINUTEN(Zeit)

(Englisch: MINUTES)

Diese Funktion extrahiert aus Zeit den Minutenwert. Die Stundenzahl wird dabei eingerechnet.

Beispiele:

MINUTEN("15:23:02") ergibt 923

Verwandte Funktionen:

SEKUNDE, SEKUNDEN, MINUTE, STUNDE, STUNDEN

1.119 MONAT(Datum)

MONAT(Datum)

(Englisch: MONTH)

Die Funktion MONAT liefert von einem Datum die Monatszahl als Zahl zurück.

Beispiele:

MONAT(WERT("23-3-93")) ergibt 3

MONAT(2000) = 6

Verwandte Funktionen:

JAHR, TAG, WOCHENTAG, HEUTE, JETZT

1.120 MONATE(Datum)

MONATE(Datum)

(Englisch: MONTHS)

Diese Funktion berechnet die Anzahl der Monate sein Januar 1900 (d.h. Januar 1900 ergibt 0)

Beispiele:

MONATE("Sep 96") ergibt 1160

Verwandte Funktionen:

TAG, JAHR, MONAT

1.121 MONATTEXT(Datum[;Abkürzung])

MONATTEXT(Datum[;Abkürzung])

(Englisch: MONTHTEXT)

Diese Funktion gibt als Text den Monat des angegebenen Datums zurück.

Datum: Das Datum, für das der Monat berechnet werden soll.

Abkürzung: Falls vorhanden und ungleich 0 wird der Monatsname nicht ausgeschreiben und nur die (dreibuchstabige) Abkürzung zurückgegeben.

Hinweis: Die Monatsnamen werden automatisch in der gerade eingestellten Sprache ausgegeben, unabhängig davon um welche nationale TurboCalc-Version es sich handelt.

Beispiele:

MONATTEXT(WERT("1.4.96")) ergibt April

MONATTEXT("1.4.96";1) ergibt Apr (beachten Sie dabei, daß zur Übergabe des Datums die implizite Typumwandlung von Texten (siehe "implizite Typumwandlung") benutzt wird)

MONATTEXT(HEUTE()) liefert den Monatsnamen des aktuellen Monats zurück.

Verwandte Funktionen:

TAGTEXT

1.122 SEKUNDE(Zeit)

SEKUNDE(Zeit)

(Englisch: SECOND)

Diese Funktion extrahiert aus Zeit den Sekundenwert. Der Stunden- sowie Minutenwert wird dabei ignoriert.

Beispiele:

SEKUNDE("15:23:42") ergibt 42

Verwandte Funktionen:

SEKUNDEN, MINUTE, MINUTEN, STUNDE, STUNDEN

1.123 SEKUNDEN(Zeit)

SEKUNDEN(Zeit)

(Englisch: SECONDS)

Diese Funktion extrahiert aus Zeit den Sekundenwert. Der Stunden- sowie Minutenwert wird dabei eingerechnet.

Beispiele:

SEKUNDEN("15:23:42") ergibt 55422

Verwandte Funktionen:

SEKUNDE, MINUTE, MINUTEN, STUNDE, STUNDEN

1.124 STUNDE(Zeit)

STUNDE(Zeit)

(Englisch: HOUR)

Diese Funktion extrahiert aus Zeit den Stundenwert. Bei einer Stundenzahl größer als 24 (d.h. mehrere Tage) wird die Stundenzahl des letzten Tages (d.h. Stundenzahl modulo 24) zurückgegeben.

Beispiele:

STUNDE("15:03:02") ergibt 15

STUNDE("29:00:00") ergibt 5

Verwandte Funktionen:

SEKUNDE, SEKUNDEN, MINUTE, MINUTEN, STUNDEN

1.125 STUNDEN(Zeit)

STUNDEN(Zeit)

(Englisch: HOURS)

Diese Funktion extrahiert aus Zeit den Stundenwert. Stundenzahlen größer als 24 (d.h. mehrere Tage) sind möglich.

Beispiele:

STUNDEN("15:03:02") ergibt 15

STUNDEN("29:00:00") ergibt 29

Verwandte Funktionen:

SEKUNDE, SEKUNDEN, MINUTE, MINUTEN, STUNDE

1.126 TAG(Datum)

TAG(Datum)

(Englisch: DAY)

Die Funktion TAG liefert von einem Datum die Tageszahl als Zahl zurück.

Beispiele:

TAG(WERT("23-3-93")) ergibt 23

TAG(2000) = 24

Verwandte Funktionen:

MONAT, JAHR, WOCHENTAG, HEUTE, JETZT

1.127 TAGTEXT(Datum[;Abkürzung])

TAGTEXT(Datum[;Abkürzung])

(Englisch: WEEKDAYTEXT)

Diese Funktion gibt als Text den Tag des angegebenen Datums zurück.

Datum: Das Datum, für das der Tag berechnet werden soll.

Abkürzung: Falls vorhanden und ungleich 0 wird der Tag nicht ausgeschrieben sondern nur die Abkürzung zurückgegeben.

Hinweis: Die Namen werden automatisch in der gerade eingestellten Sprache ausgegeben, unabhängig davon um welche nationale TurboCalc-Version es sich handelt.

Beispiele:

TAGTEXT(WERT("1.4.96")) ergibt Donnerstag.

TAGTEXT("1.4.96";1) ergibt Do (beachten Sie dabei, daß zur Übergabe des Datums die implizite Typumwandlung von Texten (siehe "implizite Typumwandlung") benutzt wird)

TAGTEXT(HEUTE()) liefert den aktuellen Wochentag zurück.

Verwandte Funktionen:

MONATTEXT

1.128 WOCHENTAG(Datum)

WOCHENTAG(Datum)

(Englisch: WEEKDAY)

Die Funktion WOCHENTAG berechnet aus einem Datum den Wochentag dieses Datums und gibt ihn als Zahl von 1 bis 7 zurück (1 = Sonntag, 7 = Samstag)

Beispiele:

WOCHENTAG(DATUM("1 Juni 92")) ergibt 3 (Dienstag)

AUSWAHL(WOCHENTAG(HEUTE);"So";"Mo";"Di";"Mi";"Do";"Fr";"Sa") ergibt das zweistellige Kürzel des heutigen Tages.

Verwandte Funktionen:

JAHR, MONAT, TAG, HEUTE, JETZT

1.129 ZEIT(Stunde;Minute;Sekunde)

ZEIT(Stunde;Minute;Sekunde)

(Englisch: TIME)

Die Funktion ZEIT wandelt die per Stunde, Minute und Sekunde angegebene Zeit in eine Uhrzeit um.

Stunde ist eine ganze Zahl von 0 bis 23, die die Stunde bestimmt.

Minute ist eine ganze Zahl von 0 bis 59, die die Minute bestimmt.

Sekunde ist eine ganze Zahl von 0 bis 59, die die Sekunde bestimmt.

Beispiele:

ZEIT(13;45;0) ergibt 13 Uhr 45 und 0 Sekunden.

Verwandte Funktionen:

WERT, DATUM, HEUTE, DATWERT, ZEITWERT

1.130 ZEITWERT(Text)

ZEITWERT(Text)

(Englisch: TIMEVALUE)

Die Funktion ZEITWERT wandelt einen Text in eine Uhrzeit um. (Es kann auch einfach WERT benutzt werden, was alle Formate umwandeln kann, siehe dort.)

Text kann ein beliebiges Zeitformat in Textform sein, also:

13:03 oder

13:03:05 (mit Sekundenangabe)

Tip: Diese Funktion ist sehr praktisch, wenn man Zeitwerte in Formeln verwenden möchte, da man obige Daten nicht einfach direkt eingeben kann.

Beispiele:

ZEITWERT("13:45") ergibt 13 Uhr 45 und 0 Sekunden.

Verwandte Funktionen:

WERT, DATUM, HEUTE, DATWERT, ZEIT

1.131 Tabellenfunktionen

Tabellenfunktionen

ANZAHL(Bereich[:])

ANZAHL2(Bereich[:])

ERSTER(Bereich;...)

LETZTER(Bereich;...)

MAX(Bereich[:])

MIN(Bereich[:])

MITTELWERT(Bereich[:])

PRODUKT(Bereich[:])

STABW(Bereich[:])

SUMME(Bereich[:])

VANZAHL(Bereich[:])

VANZAHL2(Bereich[:])

VARIANZ(Bereich[:])

VERSTER(Bereich;...)

VLETZTER(Bereich;...)

VMAX(Bereich[:])

VMIN(Bereich[:])

VMITTELWERT(Bereich[:])

VPRODUKT(Bereich[:])

VSTABW(Bereich[:])

VSUMME(Bereich[:])

VVARIANZ(Bereich[;])

VWÄHLEN(Index;Bereich;..)

WÄHLEN(Index;Bereich;...)

Die Tabellenfunktionen haben als Parameter alle eine Liste aus Bereichen.

Ein Bereich ist dabei entweder eine Zelle (also etwa A1) oder aber ein rechteckiger Zellblock, der durch Zelle1 + ":" + Zelle2 angegeben wird (z.B. A1:C3). Statt einem Doppelpunkt kann zur Kompatibilität auch ".." (zwei Punkte nacheinander) verwandt werden.

Natürlich können die Zellen/Bereiche auch absolut sein (mit Dollar "\$") oder aber durch Namen ersetzt werden, siehe dazu bei "Namen".

Für alle unten stehenden Funktionen können beliebig viele Bereiche, jeweils durch Semikolon getrennt, nacheinander aufgeführt werden. Weiterhin können auch direkt Zahlen als Parameter eingegeben werden; dies ist vor allem bei MIN und MAX sinnvoll, Beispiele siehe dort.

Hinweis: Bei den Funktionen, denen ein "V" (wie Verborgen) vorangestellt ist (außer VARIANZ), werden nur die sichtbaren Zellen berücksichtigt, d.h. Zellen die nicht ausgeblendet wurden (z.B. durch Gliederung oder Ausblenden).

Für alle Beispiele wird jeweils folgende Tabelle benutzt:

Zeile/Spalte A B C

1 2 4

2 3 5

3 Text

4

5

1.132 ANZAHL(Bereich[;])

ANZAHL(Bereich[;])

(Englisch: COUNT)

Die Funktion ANZAHL liefert die Anzahl der Zahlen in den Bereichen. Texte und leere Zellen werden ignoriert.

Beispiele:

ANZAHL(A1:A5) = 2

ANZAHL(A1;A2:B2;A3:B3) = 4

Verwandte Funktionen:

ANZAHL2, MAX, MIN, SUMME, PRODUKT

1.133 ANZAHL2(Bereich[;])

ANZAHL2(Bereich[;])

(Englisch: COUNT2)

Die Funktion ANZAHL2 liefert die Anzahl der nichtleeren Zellen in den angegebenen Bereichen. Leerzellen werden ignoriert, Texte, Daten sowie Uhrzeiten werden jedoch gezählt.

Beispiele:

ANZAHL(A1:A5) = 3

ANZAHL(A1;A2:B2;A3:B3) = 4

Verwandte Funktionen:

ANZAHL, MAX, MIN, SUMME, PRODUKT

1.134 ERSTER(Bereich;...)

ERSTER(Bereich;...)

(Englisch: FIRST)

Diese Funktion liefert den ersten Wert (bzw. den Verweis auf die erste Zelle) zurück, leere Zellen werden dabei ignoriert.

Handelt es sich beim Ergebnis um eine Zelle (d.h. war es kein direkt angegebener Wert), so wird ggf. diese Zelle zurückgegeben. Dieser Befehl kann somit als Parameter für Befehle eingesetzt werden, die eine Zelle oder einen Bereich als Parameter erwarten (etwa BLOCKX). Ansonsten wird der Zellinhalt als Ergebnis geliefert.

Beispiele:

ERSTER(A1:A5) = 2

ERSTER(5;A1:A5) = 5

Verwandte Funktionen:

LETZTER, WÄHLEN, ANZAHL, ANZAHL2, MIN, SUMME, PRODUKT

1.135 LETZTER(Bereich;...)

LETZTER(Bereich;...)

(Englisch: LAST)

Diese Funktion liefert den letzten Wert (bzw. den Verweis auf die erste Zelle) zurück, leere Zellen werden dabei ignoriert.

Handelt es sich beim Ergebnis um eine Zelle (d.h. war es kein direkt angegebener Wert), so wird ggf. diese Zelle zurückgegeben. Dieser Befehl kann somit als Parameter für Befehle eingesetzt werden, die eine Zelle oder einen Bereich als Parameter erwarten (etwa BLOCKX). Ansonsten wird der Zellinhalt als Ergebnis geliefert.

Beispiele:

LETZTER(A1:A5) = Text.

BLOCKX(LETZTER(A1:A5)) ergibt 1 (Zelle A3)

Verwandte Funktionen:

ERSTER, WÄHLEN, ANZAHL, ANZAHL2, MIN, SUMME, PRODUKT

1.136 MAX(Bereich[;])

MAX(Bereich[;])

Die Funktion MAX liefert die größte Zahl der angegebenen Bereiche. Texte und leere Zellen werden ignoriert. Enthält der Bereich keine Zahlen, so wird 0 zurückgegeben.

Beispiele:

MAX(A1:A5) = 3

MAX(A1;A2:B2;A3:B3) = 5

MAX(A1:F20;1) sucht das Maximum aus dem angegebenen Bereich. Ist es kleiner als 1, so wird 1 verwandt.

Verwandte Funktionen:

ANZAHL, ANZAHL2, MIN, SUMME, PRODUKT

1.137 MIN(Bereich[;])

MIN(Bereich[;])

Die Funktion MIN liefert die kleinste Zahl der angegebenen Bereiche. Texte und leere Zellen werden ignoriert. Enthält der Bereich keine Zahlen, so wird 0 zurückgegeben.

Beispiele:

MIN(A1:A5) = 1

MIN(A1;A2:B2;A3:B3) = 1

MIN(A1;A4) = 0

MIN(A1:F20;100) bestimmt das Minimum des angegebenen Bereichs. Ist dieses größer als 100, so wird 100 verwandt.

Verwandte Funktionen:

ANZAHL, ANZAHL2, MAX, SUMME, PRODUKT

1.138 MITTELWERT(Bereich[;])

MITTELWERT(Bereich[;])

(Englisch: AVERAGE)

Die Funktion MITTELWERT berechnet den Durchschnitt aller Zahlen der angegebenen Bereiche. Texte und leere Zellen werden ignoriert. Enthält der Bereich keine Zahlen, so wird 0 zurückgegeben.

Beispiele:

MITTELWERT(A1:A5) = 2.5

MITTELWERT(A1;A2:B2;A3:B3) = 3.5

MITTELWERT(A1;A4) = 0

Verwandte Funktionen:

ANZAHL, ANZAHL2, MAX, SUMME, PRODUKT

1.139 PRODUKT(Bereich[;])

PRODUKT(Bereich[;])

(Englisch: PRODUCT)

Die Funktion PRODUKT liefert das Produkt aller Zahlen der angegebenen Bereiche. Texte und leere Zellen werden ignoriert. Enthält der Bereich keine Zahlen, so wird 0 zurückgegeben - ist eine der Zahlen 0, so wird natürlich auch das Produkt 0.

Beispiele:

PRODUKT(A1:A5) = 6

PRODUKT(A1;A2:B2;A3:B3) = 120

Verwandte Funktionen:

ANZAHL, ANZAHL2, MAX, MIN, SUMME

1.140 STABW(Bereich[;])

STABW(Bereich[;])

(Englisch: STDEV)

Die Funktion STABW berechnet die Standardabweichung der angegebenen Werte.

Verwandte Funktionen:

ANZAHL, ANZAHL2, MAX, MIN, PRODUKT

1.141 SUMME(Bereich[;])

SUMME(Bereich[;])

(Englisch: SUM)

Die Funktion SUMME liefert die Summe aller Zahlen der angegebenen Bereiche. Texte und leere Zellen werden ignoriert. Enthält der Bereich keine Zahlen, so wird 0 zurückgegeben.

Beispiele:

SUMME(A1:A5) = 5

SUMME(A1;A2:B2;A3:B3) = 14

Verwandte Funktionen:

ANZAHL, ANZAHL2, MAX, MIN, PRODUKT

1.142 VANZAHL(Bereich[;])

VANZAHL(Bereich[;])

(Englisch: HCOUNT)

Die Funktion VANZAHL liefert die Anzahl der Zahlen in den Bereichen. Texte und leere Zellen werden ignoriert.

Hinweis: Im Gegensatz zu der Funktion ANZAHL werden bei dieser Funktion nur sichtbare Zellen berücksichtigt, d.h. Zellen die nicht ausgeblendet wurden (z.B. durch Gliederung oder Ausblenden).

Verwandte Funktionen:

VANZAHL2, VMAX, VMIN, VSUMME, VPRODUKT

1.143 VANZAHL2(Bereich[;])

VANZAHL2(Bereich[;])

(Englisch: HCOUNT2)

Die Funktion VANZAHL2 liefert die Anzahl der nichtleeren Zellen in den angegebenen Bereichen. Leerzellen werden ignoriert, Texte, Daten sowie Uhrzeiten werden jedoch gezählt.

Hinweis: Im Gegensatz zu der Funktion ANZAHL2 werden bei dieser Funktion nur sichtbare Zellen berücksichtigt, d.h. Zellen die nicht ausgeblendet wurden (z.B. durch Gliederung oder Ausblenden).

Verwandte Funktionen:

VANZAHL, VMAX, VMIN, VSUMME, VPRODUKT

1.144 VARIANZ(Bereich[;])

VARIANZ(Bereich[;])

(Englisch: VAR)

Die Funktion VARIANZ berechnet die Varianz der angegebenen Werte.

Verwandte Funktionen:

ANZAHL, ANZAHL2, MAX, MIN, PRODUKT

1.145 VERSTER(Bereich;...)

VERSTER(Bereich;...)

(Englisch: HFIRST)

Diese Funktion liefert den ersten Wert (bzw. den Verweis auf die erste Zelle) zurück, leere Zellen werden dabei ignoriert.

Handelt es sich beim Ergebnis um eine Zelle (d.h. war es kein direkt angegebener Wert), so wird ggf. diese Zelle zurückgegeben. Dieser Befehl kann somit als Parameter für Befehle eingesetzt werden, die eine Zelle oder einen Bereich als Parameter erwarten (etwa BLOCKX). Ansonsten wird der Zellinhalt als Ergebnis geliefert.

Hinweis: Im Gegensatz zu der Funktion **ERSTER** werden bei dieser Funktion nur sichtbare Zellen berücksichtigt, d.h. Zellen die nicht ausgeblendet wurden (z.B. durch Gliederung oder Ausblenden).

Verwandte Funktionen:

VLETZTER, VWÄHLEN, VANZAHL, VANZAHL2, VMIN, VSUMME, VPRODUKT

1.146 VLETZTER(Bereich;...)

VLETZTER(Bereich;...)

(Englisch: HLAST)

Diese Funktion liefert den letzten Wert (bzw. den Verweis auf die erste Zelle) zurück, leere Zellen werden dabei ignoriert.

Handelt es sich beim Ergebnis um eine Zelle (d.h. war es kein direkt angegebener Wert), so wird ggf. diese Zelle zurückgegeben. Dieser Befehl kann somit als Parameter für Befehle eingesetzt werden, die eine Zelle oder einen Bereich als Parameter erwarten (etwa BLOCKX). Ansonsten wird der Zellinhalt als Ergebnis geliefert.

Hinweis: Im Gegensatz zu der Funktion **LETZTER** werden bei dieser Funktion nur sichtbare Zellen berücksichtigt, d.h. Zellen die nicht ausgeblendet wurden (z.B. durch Gliederung oder Ausblenden).

Verwandte Funktionen:

VERSTER, VWÄHLEN, VANZAHL, VANZAHL2, VMIN, VSUMME, VPRODUKT

1.147 VMAX(Bereich[;])

VMAX(Bereich[;])

Die Funktion VMAX liefert die größte Zahl der angegebenen Bereiche. Texte und leere Zellen werden ignoriert. Enthält der Bereich keine Zahlen, so wird 0 zurückgegeben.

Hinweis: Im Gegensatz zu der Funktion **MAX** werden bei dieser Funktion nur sichtbare Zellen berücksichtigt, d.h. Zellen die nicht ausgeblendet wurden (z.B. durch Gliederung oder Ausblenden).

Verwandte Funktionen:

VANZAHL, VANZAHL2, VMIN, VSUMME, VPRODUKT

1.148 VMIN(Bereich[;])

VMIN(Bereich[;])

Die Funktion VMIN liefert die kleinste Zahl der angegebenen Bereiche. Texte und leere Zellen werden ignoriert. Enthält der Bereich keine Zahlen, so wird 0 zurückgegeben.

Hinweis: Im Gegensatz zu der Funktion **MIN** werden bei dieser Funktion nur sichtbare Zellen berücksichtigt, d.h. Zellen die nicht ausgeblendet wurden (z.B. durch Gliederung oder Ausblenden).

Verwandte Funktionen:

VANZAHL, VANZAHL2, VMAX, VSUMME, VPRODUKT

1.149 VMITTELWERT(Bereich[;])

VMITTELWERT(Bereich[;])

(Englisch: HAVERAGE)

Die Funktion VMITTELWERT berechnet den Durchschnitt aller Zahlen der angegebenen Bereiche. Texte und leere Zellen werden ignoriert. Enthält der Bereich keine Zahlen, so wird 0 zurückgegeben.

Hinweis: Im Gegensatz zu der Funktion **MITTELWERT** werden bei dieser Funktion nur sichtbare Zellen berücksichtigt, d.h. Zellen die nicht ausgeblendet wurden (z.B. durch Gliederung oder Ausblenden).

Verwandte Funktionen:

VANZAHL, VANZAHL2, VMAX, VSUMME, VPRODUKT

1.150 VPRODUKT(Bereich[;])

VPRODUKT(Bereich[;])

(Englisch: HPRODUCT)

Die Funktion VPRODUKT liefert das Produkt aller Zahlen der angegebenen Bereiche. Texte und leere Zellen werden ignoriert. Enthält der Bereich keine Zahlen, so wird 0 zurückgegeben - ist eine der Zahlen 0, so wird natürlich auch das Produkt 0.

Hinweis: Im Gegensatz zu der Funktion **PRODUKT** werden bei dieser Funktion nur sichtbare Zellen berücksichtigt, d.h. Zellen die nicht ausgeblendet wurden (z.B. durch Gliederung oder Ausblenden).

Verwandte Funktionen:

VANZAHL, VANZAHL2, VMAX, VMIN, VSUMME

1.151 VSTABW(Bereich[;])

VSTABW(Bereich[;])

(Englisch: HSTDEV)

Die Funktion VSTABW berechnet die Standardabweichung der angegebenen Werte.

Hinweis: Im Gegensatz zu der Funktion **STABW** werden bei dieser Funktion nur sichtbare Zellen berücksichtigt, d.h. Zellen die nicht ausgeblendet wurden (z.B. durch Gliederung oder Ausblenden).

Verwandte Funktionen:

VANZAHL, VANZAHL2, VMAX, VMIN, VPRODUKT

1.152 VSUMME(Bereich[;])

VSUMME(Bereich[;])

(Englisch: HSUM)

Die Funktion VSUMME liefert die Summe aller Zahlen der angegebenen Bereiche. Texte und leere Zellen werden ignoriert. Enthält der Bereich keine Zahlen, so wird 0 zurückgegeben.

Hinweis: Im Gegensatz zu der Funktion SUMME werden bei dieser Funktion nur sichtbare Zellen berücksichtigt, d.h. Zellen die nicht ausgeblendet wurden (z.B. durch Gliederung oder Ausblenden).

Verwandte Funktionen:

VANZAHL, VVANZAHL2, VMAX, VMIN, VPRODUKT

1.153 VVARIANZ(Bereich[;])

VVARIANZ(Bereich[;])

(Englisch: HVAR)

Die Funktion VVARIANZ berechnet die Varianz der angegebenen Werte.

Hinweis: Im Gegensatz zu der Funktion VARIANZ werden bei dieser Funktion nur sichtbare Zellen berücksichtigt, d.h. Zellen die nicht ausgeblendet wurden (z.B. durch Gliederung oder Ausblenden).

Verwandte Funktionen:

VANZAHL, VVANZAHL2, VMAX, VMIN, VPRODUKT

1.154 VWÄHLEN(Index;Bereich;...)

VWÄHLEN(Index;Bereich;...)

(Englisch: HPICK)

Diese Funktion liefert den Index-ten Wert (bzw. den Verweis auf die erste Zelle) zurück, leere Zellen werden dabei ignoriert.

Index: eine Ganzzahl, die bestimmt, welcher Wert ausgewählt wird: 1=erster, 2=zweiter,... -1 bedeutet letzter Wert.

Handelt es sich beim Ergebnis um eine Zelle (d.h. war es kein direkt angegebener Wert), so wird ggf. diese Zelle zurückgegeben. Dieser Befehl kann somit als Parameter für Befehle eingesetzt werden, die eine Zelle oder einen Bereich als Parameter erwarten (etwa BLOCKX). Ansonsten wird der Zellinhalt als Ergebnis geliefert.

Hinweis: Im Gegensatz zu der Funktion WÄHLEN werden bei dieser Funktion nur sichtbare Zellen berücksichtigt, d.h. Zellen die nicht ausgeblendet wurden (z.B. durch Gliederung oder Ausblenden).

Verwandte Funktionen:

VERSTER, VLETZTER, VVANZAHL, VVANZAHL2, VMIN, VSUMME, VPRODUKT

1.155 WÄHLEN(Index;Bereich;...)

WÄHLEN(Index;Bereich;...)

(Englisch: PICK)

Diese Funktion liefert den Index-ten Wert (bzw. den Verweis auf die erste Zelle) zurück, leere Zellen werden dabei ignoriert.

Index: eine Ganzzahl, die bestimmt, welcher Wert ausgewählt wird: 1=erster, 2=zweiter,... -1 bedeutet letzter Wert.

Handelt es sich beim Ergebnis um eine Zelle (d.h. war es kein direkt angegebener Wert), so wird ggf. diese Zelle zurückgegeben. Dieser Befehl kann somit als Parameter für Befehle eingesetzt werden, die eine Zelle oder einen Bereich als Parameter erwarten (etwa BLOCKX). Ansonsten wird der Zellinhalt als Ergebnis geliefert.

Beispiele:

WÄHLEN(1;A1:A5) = 2

WÄHLEN(2;A1:A5) = 4

Verwandte Funktionen:

ERSTER, LETZTER, ANZAHL, ANZAHL2, MIN, SUMME, PRODUKT

1.156 Datenbankfunktionen

Datenbankfunktionen

DBANZAHL(Datenbank;Spalte;Kriterien)

DBANZAHL2(Datenbank;Spalte;Kriterien)

DBERSTER(Datenbank;Spalte;Kriterien)

DBLETZTER(Datenbank;Spalte;Kriterien)

DBMAX(Datenbank;Spalte;Kriterien)

DBMIN(Datenbank;Spalte;Kriterien)

DBMITTELWERT(Datenbank;Spalte;Kriterien)

DBPRODUKT(Datenbank;Spalte;Kriterien)

DBSTABW(Datenbank;Spalte;Kriterien)

DBSUMME(Datenbank;Spalte;Kriterien)

DBVARIANZ(Datenbank;Spalte;Kriterien)

DBWÄHLEN(Index;Datenbank;Spalte;Kriterien)

Diese Funktionen sind ähnlich den obigen Tabellenfunktionen, jedoch wird jetzt statt einem "normalen" Bereich eine Datenbank zugrunde gelegt. Alle unten aufgeführten Funktionen haben deswegen den gleichen Syntax:

DBFunktion(Datenbankbereich;Spalte;Kriterienbereich)

Datenbankbereich: Es handelt sich um einen Bereich wie er im Abschnitt "Datenbank" beschrieben wurde (also: erste Zeile Titel, restliche Zeilen Daten). Es muß sich nicht um die aktuelle definierte Tabelle handeln (kann aber, indem der Name "DATENBANK" benutzt wird).

Spalte: Dies gibt die Spalte (des Datenbankbereichs) an, auf die sich die Funktionen (nicht der Kriterienbereich) beziehen sollen. Diese kann als Nummer (0= erste Spalte, 1 = zweite Spalte...) oder als Spaltentitel-Text (in Anführungszeichen!) eingegeben werden.

Kriterienbereich: Dieser Bereich bestimmt, welche Datensätze zur Berechnung herangezogen werden sollen. Es werden nur die Datensätze (der ausgewählten Spalte) benutzt, die mit den Kriterien übereinstimmen. Näheres (auch zum Aufbau dieses Bereichs) siehe im Abschnitt "Datenbank" unter "Suchkriterien".

Tip: Da diese Funktionen nicht nur auf den aktuellen Datenbankbereich angewandt werden können, sondern auf beliebige (richtig eingerichtete) Bereiche, kann man diese Funktionen auch als Auswahlkriterien nutzen (um etwa die Spalten mit einem Wert>2000 oder ein Datum zwischen 1.1.1990 und 1.1.1991 zu zählen bzw. aufzusummieren...)

Tip: Da (vor allem bei großen Bereichen) diese Funktionen relativ lange zum Berechnen brauchen (da sie ja alle Datensätze vergleichen müssen), empfiehlt es sich, hier die automatische Berechnung auszuschalten.

1.157 DBANZAHL(Datenbank;Spalte;Kriterien)

DBANZAHL(Datenbank;Spalte;Kriterien)

(Englisch: DCOUNT)

Die Funktion DBANZAHL liefert die Anzahl der Zahlen des Datenbankbereichs, die mit den Kriterien übereinstimmen - näheres zu den Parametern siehe Einführung zu den Datenbankfunktionen.

Beispiele:

DBANZAHL(DATENBANK;0;SUCHKRITERIEN) liefert die Anzahl der Zahlen in der ersten Spalte des Datenbankbereichs, die mit den Suchkriterien übereinstimmen.

DBANZAHL(DATENBANK;"Nummer";SUCHKRITERIEN) entsprechend, falls die erste Spalte den Titel "Nummer" hat.

Verwandte Funktionen:

[Datenbankfunktionen](#) und [Tabellenfunktionen](#)

1.158 DBANZAHL2(Datenbank;Spalte;Kriterien)

DBANZAHL2(Datenbank;Spalte;Kriterien)

(Englisch: DCOUNT2)

Die Funktion DBANZAHL liefert die Anzahl der nichtleeren Zellen des Datenbankbereichs, die mit den Kriterien übereinstimmen - näheres zu den Parametern siehe Einführung zu den Datenbankfunktionen.

Leerzellen werden ignoriert, Texte, Daten sowie Uhrzeiten werden jedoch gezählt.

Diese Funktion ist sehr nützlich zum Bestimmen der Anzahl der Datensätze, die den Suchkriterien entsprechen.

Beispiele:

DBANZAHL2(DATENBANK;0;SUCHKRITERIEN) liefert die Anzahl der nichtleeren Zellen in der ersten Spalte des Datenbankbereichs, die mit den Suchkriterien übereinstimmen.

DBANZAHL2(DATENBANK;"Nummer";SUCHKRITERIEN) entsprechend, falls die erste Spalte den Titel "Nummer" hat.

Verwandte Funktionen:

[Datenbankfunktionen](#) und [Tabellenfunktionen](#)

1.159 DBERSTER(Datenbank;Spalte;Kriterien)

DBERSTER(Datenbank;Spalte;Kriterien)

(Englisch: DFIRST)

Diese Funktion liefert den Inhalt (bzw. die Zelle) des ersten Wertes der ersten Zeile des Datenbankbereichs, die mit den Kriterien übereinstimmen - näheres zu den Parametern siehe Einführung zu den Datenbankfunktionen.

Stellen die Suchkriterien keine Übereinstimmung fest, so wird ein Fehlerwert zurückgegeben.

Dieser Befehl kann als Parameter für Befehle eingesetzt werden, die eine Zelle oder einen Bereich als Parameter erwarten (etwa BLOCKX). In diesem Fall wird der entsprechende Zellbezug zurückgegeben, sonst der Zellinhalt.

Verwandte Funktionen:

[Datenbankfunktionen](#) und [Tabellenfunktionen](#)

1.160 DBLETZTER(Datenbank;Spalte;Kriterien)

DBLETZTER(Datenbank;Spalte;Kriterien)

(Englisch: DLAST)

Diese Funktion liefert den Inhalt (bzw. die Zelle) des ersten Wertes der letzten Zeile des Datenbankbereichs, die mit den Kriterien übereinstimmen - näheres zu den Parametern siehe Einführung zu den Datenbankfunktionen.

Stellen die Suchkriterien keine Übereinstimmung fest, so wird ein Fehlerwert zurückgegeben.

Dieser Befehl kann als Parameter für Befehle eingesetzt werden, die eine Zelle oder einen Bereich als Parameter erwarten (etwa BLOCKX). In diesem Fall wird der entsprechende Zellbezug zurückgegeben, sonst der Zellinhalt.

Verwandte Funktionen:

Datenbankfunktionen und **Tabellenfunktionen**

1.161 DBMAX(Datenbank;Spalte;Kriterien)

DBMAX(Datenbank;Spalte;Kriterien)

(Englisch: DMAX)

Die Funktion DBMAX liefert die größte Zahl der gewünschten Spalte aller Zeilen des Datenbankbereichs, die mit den Kriterien übereinstimmen - näheres zu den Parametern siehe Einführung zu den Datenbankfunktionen.

Stellen die Suchkriterien keine Übereinstimmung fest bzw. sind in diesen Zeilen in der ausgewählten Spalte keine Zahlen, so wird 0 zurückgegeben.

Beispiele:

DBMAX(DATENBANK;0;SUCHKRITERIEN) liefert die größte Zahl der ersten Spalte des Datenbankbereichs, die mit den Suchkriterien übereinstimmen.

DBMAX(DATENBANK;"Nummer";SUCHKRITERIEN) entsprechend, falls die erste Spalte den Titel "Nummer" hat.

Verwandte Funktionen:

Datenbankfunktionen und **Tabellenfunktionen**

1.162 DBMIN(Datenbank;Spalte;Kriterien)

DBMIN(Datenbank;Spalte;Kriterien)

(Englisch: DMIN)

Die Funktion DBMIN liefert die kleinste Zahl der gewünschten Spalte aller Zeilen des Datenbankbereichs, die mit den Kriterien übereinstimmen - näheres zu den Parametern siehe Einführung zu den Datenbankfunktionen.

Stellen die Suchkriterien keine Übereinstimmung fest bzw. sind in diesen Zeilen in der ausgewählten Spalte keine Zahlen, so wird 0 zurückgegeben.

Beispiele:

DBMIN(DATENBANK;0;SUCHKRITERIEN) liefert die kleinste Zahl der ersten Spalte des Datenbankbereichs, die mit den Suchkriterien übereinstimmen.

DBMIN(DATENBANK;"Nummer";SUCHKRITERIEN) entsprechend, falls die erste Spalte den Titel "Nummer" hat.

Verwandte Funktionen:

Datenbankfunktionen und **Tabellenfunktionen**

1.163 DBMITTELWERT(Datenbank;Spalte;Kriterien)

DBMITTELWERT(Datenbank;Spalte;Kriterien)

(Englisch: DAVERAGE)

Die Funktion DBMITTELWERT liefert den Mittelwert aller Zahlen der gewünschten Spalte aller Zeilen des Datenbankbereichs, die mit den Kriterien übereinstimmen - näheres zu den Parametern siehe Einführung zu den Datenbankfunktionen.

Stellen die Suchkriterien keine Übereinstimmung fest bzw. sind in diesen Zeilen in der ausgewählten Spalte keine Zahlen, so wird 0 zurückgegeben.

Beispiele:

DBMITTELWERT(DATENBANK;0;SUCHKRITERIEN) liefert den Durchschnitt der ersten Spalte des Datenbankbereichs, die mit den Suchkriterien übereinstimmen.

DBMITTELWERT(DATENBANK;"Nummer";SUCHKRITERIEN) entsprechend, falls die erste Spalte den Titel "Nummer" hat.

Verwandte Funktionen:

[Datenbankfunktionen](#) und [Tabellenfunktionen](#)

1.164 DBPRODUKT(Datenbank;Spalte;Kriterien)

DBPRODUKT(Datenbank;Spalte;Kriterien)

(Englisch: DPRODUCT)

Die Funktion DBPRODUKT liefert das Produkt aller Zahlen der gewünschten Spalte aller Zeilen des Datenbankbereichs, die mit den Kriterien übereinstimmen - näheres zu den Parametern siehe Einführung zu den Datenbankfunktionen.

Stellen die Suchkriterien keine Übereinstimmung fest bzw. sind in diesen Zeilen in der ausgewählten Spalte keine Zahlen, so wird 0 zurückgegeben.

Beispiele:

DBPRODUKT(DATENBANK;0;SUCHKRITERIEN) multipliziert alle Zahlen der ersten Spalte des Datenbankbereichs, die mit den Suchkriterien übereinstimmen.

DBPRODUKT(DATENBANK;"Nummer";SUCHKRITERIEN) entsprechend, falls die erste Spalte den Titel "Nummer" hat.

Verwandte Funktionen:

[Datenbankfunktionen](#) und [Tabellenfunktionen](#)

1.165 DBSTABW(Datenbank;Spalte;Kriterien)

DBSTABW(Datenbank;Spalte;Kriterien)

(Englisch: DSTDEV)

Die Funktion DBSTABW berechnet die Standardabweichung aller Zahlen der gewünschten Spalte aller Zeilen des Datenbankbereichs, die mit den Kriterien übereinstimmen - näheres zu den Parametern siehe Einführung zu den Datenbankfunktionen.

Verwandte Funktionen:

[Datenbankfunktionen](#) und [Tabellenfunktionen](#)

1.166 DBSUMME(Datenbank;Spalte;Kriterien)

DBSUMME(Datenbank;Spalte;Kriterien)

(Englisch: DSUM)

Die Funktion DBSUMME liefert die Summe aller Zahlen der gewünschten Spalte aller Zeilen des Datenbankbereichs, die mit den Kriterien übereinstimmen - näheres zu den Parametern siehe Einführung zu den Datenbankfunktionen.

Stellen die Suchkriterien keine Übereinstimmung fest bzw. sind in diesen Zeilen in der ausgewählten Spalte keine Zahlen, so wird 0 zurückgegeben.

Beispiele:

DBSUMME(DATENBANK;0;SUCHKRITERIEN) addiert alle Zahlen der ersten Spalte des Datenbankbereichs, die mit den Suchkriterien übereinstimmen.

DBSUMME(DATENBANK;"Nummer";SUCHKRITERIEN) entsprechend, falls die erste Spalte den Titel "Nummer" hat.

Verwandte Funktionen:

[Datenbankfunktionen](#) und [Tabellenfunktionen](#)

1.167 DBVARIANZ(Datenbank;Spalte;Kriterien)

DBVARIANZ(Datenbank;Spalte;Kriterien)

(Englisch: DVAR)

Die Funktion DBVARIANZ berechnet die Varianz aller Zahlen der gewünschten Spalte aller Zeilen des Datenbankbereichs, die mit den Kriterien übereinstimmen - näheres zu den Parametern siehe Einführung zu den Datenbankfunktionen.

Verwandte Funktionen:

[Datenbankfunktionen](#) und [Tabellenfunktionen](#)

1.168 DBWÄHLEN(Index;Datenbank;Spalte;Kriterien)

DBWÄHLEN(Index;Datenbank;Spalte;Kriterien)

(Englisch: DPICK)

Diese Funktion liefert den Inhalt (bzw. die Zelle) des ersten Wertes der Index-ten Zeilen des Datenbankbereichs, die mit den Kriterien übereinstimmen - näheres zu den Parametern siehe Einführung zu den Datenbankfunktionen.

Stellen die Suchkriterien keine Übereinstimmung fest, so wird ein Fehlerwert zurückgegeben.

Dieser Befehl kann als Parameter für Befehle eingesetzt werden, die eine Zelle oder einen Bereich als Parameter erwarten (etwa BLOCKX). In diesem Fall wird der entsprechende Zellbezug zurückgegeben, sonst der Zellinhalt.

Verwandte Funktionen:

[Datenbankfunktionen](#) und [Tabellenfunktionen](#)

1.169 Zellfunktionen

Zellfunktionen

[#Bezug](#)

[@Tabelle;Bezug](#)

AKTUELLERBEREICH()
AUSWAHL(Index; Wert1; Wert2; Wert3)
BEREICHABS(Zeile;Spalte;Höhe;Breite)
BLATTNAME()
BLOCKBREITE([Bereich])
BLOCKHÖHE([Bereich])
BLOCKX([Bereich])
BLOCKY([Bereich])
INDIREKT(Text)
MAPPENNAME()
SPALTENNUMMER([Bereich])
SVERWEIS(Wert;Bereich;Offset[;Exakt])
TABELLENNAME()
VERWEIS(Wert;Bereich[;Exakt])
WVERWEIS(Wert;Bereich;Offset[;Exakt])
ZEILENUMMER([Bereich])
ZELLE(Zeile;Spalte)
ZELLEABS(Zeile;Spalte)

1.170 #Bezug

#Bezug

(Diese Funktion ist nur in Makrobefehlen definiert, ansonsten wird sie meist eine Fehlermeldung oder einen beliebigen Wert anzeigen).

Bezüge innerhalb von Makrobefehlen beziehen sich fast immer auf die Tabelle, aus der das Makro aufgerufen wurde und nicht auf die Makrotabellen (Ausnahmen: SPRINGE, siehe dort).

Will man eine Zelle der Makrotabelle ansprechen, so ist dies mit #Bezug möglich, wobei Bezug die gewünschte Zelle bzw. den gewünschten Bereich darstellt.

(#Bezug entspricht also @Makrotabelle;Bezug, wenn Makrotabelle der Tabellename ist, in der sich das gerade laufende Makro befindet).

Beispiele:

folgender Makroausschnitt realisiert eine Sicherheitsabfrage:

A10 =ANFRAGE("Ja oder Nein?")

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

A12 hier steht das eigentliche Programm

... ..

A20 =RETURN()

Der erste Befehl ANFRAGE fragt nach (hier kurz: "Ja oder Nein?"), ob etwas ausgeführt werden soll.

Der Anwender kann dann auf >OK< oder >Abbruch< klicken. Bei >OK< befindet sich WAHR in Zelle A10, sonst FALSCH

In Zelle A11 wird dann dieser Inhalt geprüft. (Würde hier das # fehlen und wurde das Makro aus einer anderen Tabelle heraus gestartet, so würde der Inhalt der Zelle dieser Tabelle ausgelesen - und dies kann ja beliebig sein!). Ist er FALSCH, dann ist not(#A10) WAHR und so wird zu Zelle A20 gesprungen, also die eigentliche Routine ausgelassen.

Verwandte Funktionen:

@, AT

1.171 @Tabelle;Bezug

@Tabelle;Bezug

(oder AT(Tabelle;Bezug))

Die AT-Funktion (d.h. der "Klammeraffe" (Alt-2)) liest Werte aus Zellen von anderen Tabellen aus, kann also benutzt werden, um mehrere Tabellen miteinander zu verknüpfen.

Tabelle: Dies ist der Name einer Tabelle (wie er als Fenster-Titel erscheint). Es die Datei geladen, so reicht der Name ohne Pfadangabe und Endung, sofern diese .TCD ist. Der Name wird mit Semikolon ";" abgeschlossen. Der Name kann in Anführungszeichen eingeschlossen werden (und muß es, falls er ein Semikolon ";" enthält!) Ist die Tabelle nicht geladen, so muß eine Pfadangabe mit angegeben werden, damit TurboCalc die Datei finden kann (siehe auch unten).

Die geladene Datei "Dh0:Daten/Test.TCD" kann also mit Test, Test.TCD, Daten/Test.TCD oder dh0:Daten/Test.TCD angesprochen werden.

Bezug: Gibt einen normalen Bezug (etwa A1 oder \$A\$1). Es spielt dabei keine Rolle, ob der Bezug relativ oder absolut (d.h. mit Dollarzeichen) angegeben wird.

Hinweis: Es kann wahlweise @ oder AT benutzt werden, weiterhin die beiden (durch Semikolon getrennten Parameter!) auch geklammert werden.

Ab der Version 3 kann mit der AT-Funktion auch auf geschlossene Tabellen zugegriffen werden: (Jedoch nur zum Auslesen von Zellen, etwa @Test;a7 - für Bereichs-Angaben mit diesem Befehl muß die andere Tabelle weiterhin geöffnet.).

Ist jedoch nun die gewünschte Tabelle nicht geöffnet, so wird versucht, die angegebene Zelle aus der Datei auszulesen. Das gelesene Ergebnis wird intern zwischengespeichert, so daß nur einmal auf Disk zugegriffen wird. Die Suchreihenfolge ist wie folgt:

i) Enthält der Dateinamen einen Doppelpunkt ":", so wird nur in diesem Pfad gesucht (da es sich ja um eine "absoluten" Pfadangabe handelt).

ii) Ansonsten wird nacheinander in folgenden Pfaden gesucht:

1. Pfad der Tabelle, in der die Funktion eingegeben wurde. (Falls hier ein Pfad gesetzt ist, d.h. im Dateinamen ein ":" oder "/" vorhanden ist).
2. Pfad aus dem eine Tabelle per WorkBench gestartet wurde (Doppelklick auf Icon oder Shift-Klick auf Icon und Programm)
3. Pfad, in dem sich TurboCalc befindet.
4. Aktiver Pfad beim Start von TurboCalc (nur Shell)

Konnte die Datei bis jetzt nicht gefunden werden, so wird ein ".TCD" angehängt und die Suche von 1. bis 4. wiederholt.

Daher: Die Angabe von .TCD ist nicht notwendig. Es empfiehlt sich jedoch, den Pfad komplett anzugeben (und ggf. auch .TCD). Um so sicherer kann TurboCalc die Tabelle finden. Aber: Dadurch wird auch ein Portieren erschwert. Je nach Anwendung ist daher auch eine "relative" Angabe (zur aktuellen Tabelle) sinnvoll.

Tip: Der gelesene Wert (oder Text) wird intern mit Tabellennamen und Zelle zwischen gespeichert, so daß er nur einmal gelesen werden muß, aber: hierbei wird zwischen verschiedenen Namen unterschieden (mit/ohne ".TCD", anderer Pfad) - auch wenn alle Namen möglicherweise zu den gleichen Daten führen! (Groß/Kleinschreibung ist egal) Deshalb: Möglichst immer die gleiche Schreibweise benutzen (etwa einfach durch Kopieren der Formel). Ansonsten kann es passieren, daß die gleichen Daten mehrfach gelesen und zwischen gespeichert werden. (Dies führt natürlich zu keinen Fehler, erhöht jedoch den Speicherverbrauch.)

Hinweis: Konnte die Tabelle nicht geöffnet werden, so erscheint weiterhin eine Fehlermeldung (ist die Zelle leer, so wird 0 zurückgegeben.)

Anmerkung: Mit Paßwort geschützte Zellen können hiermit nicht angesprochen werden (diese müssen geöffnet sein). Diese werden als "nicht gefunden" interpretiert.

WICHTIGER HINWEIS: In den älteren Version von TC2 (bis TC2.18) enthielt die Speicherroutine einen kleinen Fehler, der ein Einlesen von Werten aus so gespeicherten Tabellen verhinderte (sobald Formeln gespeichert wurden). Einfache Abhilfe: Tabelle(n) laden und mit der aktuellen Version wieder Speichern.

Beispiele:

Wenn die Datei DH0:Daten/Test.TCD geöffnet ist und in der Zelle C5 der Wert 123 steht, so liefern alle folgenden Ausdrücke diesen Wert zurück:

@Test;C5

@ "Test";\$C\$5

@ "Daten/Test";C\$5

@Daten/Test.TCD;C5

@ "DH0:daten/test";C5

@ ("Test";C5)

AT(Test;C5)

Lautet der Dateiname z.B. "Abrechnung Januar; Februar.TCD", so müssen Anführungszeichen stehen:

@ "Abrechnung Januar; Februar";A1 oder

@ "Abrechnung Januar; Februar.TCD";A1

Befehlserweiterung:

Für den Umgang mit Mappen wurde die Funktion so erweitert, daß damit auch auf andere Blätter der gleichen Mappe zugegriffen werden kann:

AT(Blattname;Zelle) bzw.

@ Blattname;Zelle

Blattname kann dabei der Name einer anderen Tabelle oder (ab TurboCalc 4) auch der Name eines anderen Blattes (der aktuellen Mappe) sein.

Entspricht Blattname dem Namen eines Blattes der aktuellen Mappe, so wird die gewünschte Zelle daraus gelesen, ansonsten wird versucht auf eine andere Tabelle (bzw. Mappe) zuzugreifen.

Hinweis: Somit können Sie mehrere mit TurboCalc 3 erstellte Tabellen einfach zu einer Mappe zusammenfassen, ohne die Formeln ändern zu müssen. (Ein ggf. im Blattnamen vorhandener Pfad bzw. die Endung ".TCD" muß jedoch entfernt werden!)

Zweite Variante

Um den Umgang mit anderen Blättern zu vereinfachen wurde eine alternative einfachere und intuitivere Schreibweise eingeführt:

Blattname@Zelle

Blattname@Bereich

Hinweise:

- * Wie die zweite Zeile zeigt, kann hiermit auch auf einen Bereich (A1:C5) zugegriffen werden.
- * Die Funktion kann auch geschachtelt werden, um Blätter einer anderen Mappe zuzugreifen: Mappenname@Blattname@Zelle.
- * In dieser Variante darf Blattname keine Leerzeichen oder andere Sonderzeichen enthalten.
- * Mit dieser Variante kann nur auf geöffnete Mappen bzw. Tabellen zugegriffen werden.

Beispiele:

=WENN(Daten@A5;Daten@A7*2;0)

=SUMME(Daten@A1:C7)

=Daten@Einnahmen

=Mappe2@Blatt2@C7

=Daten@WENN(A5;A7*2;0) (hier wirkt Daten@ auf alle Zellbezüge innerhalb WENN)

Verwandte Funktionen:

#Bezug

1.172 AKTUELLERBEREICH()

AKTUELLERBEREICH()

(Englisch: CURRENTRANGE)

Diese Funktion liefert als Ergebnis den aktuellen Bereich (d.h. der in der aktuellen Tabelle markierte Bereich). Dies kann vor allem im Zusammenhang mit Makros nützlich sein.

Anmerkung: Wird diese Funktion direkt in eine Zelle eingegeben (d.h. nicht als Parameter eines Makrobefehls), so hängt das Ergebnis von der ausgewählten Zelle bzw. dem ausgewählten Bereich zum Zeitpunkt des Neuberechnens abhängig.

Beispiele:

SUMME(AKTUELLERBEREICH()) berechnet die Summe aller Werte des aktuellen Bereichs.

Verwandte Funktionen:

BLOCKX, BLOCKY, BLOCKBREITE, BLOCKHÖHE, ZEILENNUMMER, SPALTENNUMMER

1.173 AUSWAHL(Index; Wert1; Wert2; Wert3)

AUSWAHL(Index; Wert1; Wert2; Wert3)

(Englisch: CHOOSE)

Die Funktion AUSWAHL gibt in Abhängigkeit von Index den Wert1, 2... zurück.

Dabei stellt Index eine positive Zahl ab 1 dar, die angibt, welcher der darauf folgenden Werte zurückgegeben werden soll. Ist Index kleiner eins oder aber größer als die Anzahl der Werte, so wird #WERT zurückgegeben.

Wert1,2... können beliebige Ausdrücke (Texte, Zahlen...) sein, die dann zurückgegeben werden. (Auch weitere AUSWAHL(...;...;...) als Parameter sind möglich - AUSWAHL läßt sich also schachteln).

Beispiele:

AUSWAHL(1;"Hallo";"Du";123) ergibt "Hallo"

AUSWAHL(2;"Hallo";"Du";123) ergibt "Du"

AUSWAHL(3;"Hallo";"Du";123) ergibt 123

AUSWAHL(4;"Hallo";"Du";123) ergibt #WERT

AUSWAHL(WOCHENTAG(HEUTE);"So";"Mo";"Di";"Mi";"Do";"Fr";"Sa") ergibt den zweistelligen Kürzel des heutigen Tages (WOCHENTAG gibt dabei eine Zahl zwischen 1 und 7 zurück!)

Verwandte Funktionen:

WENN

1.174 BEREICHABS(Zeile;Spalte;Höhe;Breite)

BEREICHABS(Zeile;Spalte;Höhe;Breite)

(Englisch: RANGEABS)

Die Funktion BEREICHABS definiert einen Block der Breite Breite und der Höhe Höhe ab der Position Zeile, Spalte. Alle Werte müssen größer als 0 sein (bei kleineren Werten erscheint #WERT)!

Hinweis: Bitte beachten Sie, daß zuerst die Zeile und dann die Spalte kommt. Dies ist die Art, wie Tabellen normalerweise gelesen werden (entsprechend Matrizen!) - im Koordinatensystem würde dies jedoch Y,X bedeuten! Gleiches gilt dann auch für die Höhe und Breite.

Tip: Diese Funktion kann in allen Funktionen bzw. Befehlen als Parameter benutzt werden, die einen Block als Parameter erwarten. Somit lassen sich innerhalb von Makros recht einfach benutzerdefinierte Bereiche angeben.

Tip: Statt BEREICHABS kann auch ZELLEABS benutzt werden, etwa: ZELLEABS(1;3):ZELLEABS(7;5)

Beispiele:

BEREICHABS(3;4;1;2) entspricht (D3:D4).

SELECT(BEREICHABS(3;4;1;2)) wählt den entsprechenden Bereich aus.

Verwandte Funktionen:

ZELLE, INDIREKT, ZELLEABS

1.175 BLATTNAME()

BLATTNAME()

(Englisch: SHEETNAME2)

Gibt den Namen des aktuellen Blattes zurück, d.h. das Blatt, in der sich die Formel befindet bzw. (im Makromodus) das gerade aktive Blatt. Dieser Befehl kann im Zusammenhang mit dem Makrobefehl WÄHLETABELLE sinnvoll eingesetzt werden.

Beispiele:

BLATTNAME() ergibt in der Standardmappe den Text "Blatt1"

Verwandte Funktionen:

MAPPENNAME, TABELLENNAME

1.176 BLOCKBREITE([Bereich])

BLOCKBREITE([Bereich])

(Englisch: RANGEWIDTH)

Diese Funktion gibt die Breite des als Parameter angegebenen Bereichs zurück. Dies kann vor allem im Zusammenhang mit Makros ganz nützlich sein.

Wird Bereich nicht angegeben, so wird die Breite des aktuellen Blockes zurückgegeben (also der, der in der aktuellen Tabelle markiert ist). Damit kann, zusammen mit BLOCKX, BLOCKY und BLOCKHÖHE der aktuelle Block von einem Makro ausgelesen werden.

Beispiele:

BLOCKBREITE(A1:C5) ergibt 3

BLOCKBREITE(F7) ergibt 1

BLOCKBREITE() gibt die Breite des aktuellen Blocks zurück.

Verwandte Funktionen:

BLOCKX, BLOCKY, BLOCKHÖHE, ZEILENNUMMER, SPALTENNUMMER

1.177 BLOCKHÖHE([Bereich])

BLOCKHÖHE([Bereich])

(Englisch: RANGEHEIGHT)

Diese Funktion gibt die Höhe des als Parameter angegebenen Bereichs zurück. Dies kann vor allem im Zusammenhang mit Makros ganz nützlich sein.

Wird Bereich nicht angegeben, so wird die Höhe des aktuellen Blockes zurückgegeben (also der, der in der aktuellen Tabelle markiert ist). Damit kann, zusammen mit BLOCKX, BLOCKY und BLOCKBREITE der aktuelle Block von einem Makro ausgelesen werden.

Beispiele:

BLOCKHÖHE(A1:C5) ergibt 5

BLOCKHÖHE(F7) ergibt 1

BLOCKHÖHE() gibt die Höhe des aktuellen Blocks zurück.

Verwandte Funktionen:

BLOCKX, BLOCKY, BLOCKBREITE, ZEILENNUMMER, SPALTENNUMMER

1.178 BLOCKX([Bereich])

BLOCKX([Bereich])

(Englisch: RANGEX)

Diese Funktion gibt die X-Position des als Parameter angegebenen Bereichs zurück. Dies kann vor allem im Zusammenhang mit Makros ganz nützlich sein.

Wird Bereich nicht angegeben, so wird die X-Position des aktuellen Blocks zurückgegeben (also der, der in der aktuellen Tabelle markiert ist). Damit kann, zusammen mit BLOCKY, BLOCKBREITE und BLOCKHÖHE der aktuelle Block von einem Makro ausgelesen werden.

Hinweis: Diese Funktion ist ähnlich der Funktion SPALTENNUMMER, jedoch mit folgendem wichtigen Unterschied: SPALTENNUMMER bezieht sich immer auf die Cursor-Position und diese muß nicht immer mit der linken oberen Ecke des Blocks übereinstimmen!

Beispiele:

BLOCKX(A1:C5) ergibt 1

BLOCKX(F7) ergibt 5

BLOCKX() gibt die X-Position des aktuellen Blocks zurück.

Verwandte Funktionen:

BLOCKY, BLOCKBREITE, BLOCKHÖHE, ZEILENNUMMER, SPALTENNUMMER

1.179 BLOCKY([Bereich])

BLOCKY([Bereich])

(Englisch: RANGEY)

Diese Funktion gibt die Y-Position des als Parameter angegebenen Bereichs zurück. Dies kann vor allem im Zusammenhang mit Makros ganz nützlich sein.

Wird Bereich nicht angegeben, so wird die Y-Position des aktuellen Blocks zurückgegeben (also der, der in der aktuellen Tabelle markiert ist). Damit kann, zusammen mit BLOCKX, BLOCKBREITE und BLOCKHÖHE der aktuelle Block von einem Makro ausgelesen werden.

Hinweis: Diese Funktion ist ähnlich der Funktion ZEILENNUMMER, jedoch mit folgendem wichtigen Unterschied: ZEILENNUMMER bezieht sich immer auf die Cursor-Position und diese muß nicht immer mit der linken oberen Ecke des Blocks übereinstimmen!

Beispiele:

BLOCKY(A1:C5) ergibt 1

BLOCKY(F7) ergibt 7

BLOCKY() gibt die Y-Position des aktuellen Blocks zurück.

Verwandte Funktionen:

BLOCKX, **BLOCKBREITE**, **BLOCKHÖHE**, **ZEILENNUMMER**, **SPALTENNUMMER**

1.180 INDIREKT(Text)

INDIREKT(Text)

(Englisch: INDIRECT)

Die Funktion INDIREKT liest den Zellbezug aus dem Text Text aus.

Text: Ist ein Text (in Anführungszeichen!), der einen gültigen Zellbezug (bzw. Namen) enthält. (Auch die Funktionen ZELLE, ZELLEABS und INDIREKT sind im Text möglich).

Beispiele:

INDIREKT("A1") ergibt die Zelle A1

INDIREKT(A1) ergibt \$F\$5, falls in der Zelle A1 der Text "\$F\$5" steht.

Verwandte Funktionen:

ZELLE; **ZELLEABS**, **BEREICHABS**

1.181 MAPPENNAME()

MAPPENNAME()

(Englisch: FOLDERNAME)

Gibt den Namen der aktuellen Mappe zurück, d.h. die Mappe, in der sich die Formel befindet bzw. (im Makromodus) die gerade aktive Mappe. Dieser Befehl kann im Zusammenhang mit dem Makrobefehl WÄHLETABELLE sinnvoll eingesetzt werden.

Beispiele:

MAPPENNAME() ergibt in der Standardmappe den Text "Mappe1"

Verwandte Funktionen:

BLATTNAME, **TABELLENNAME**

1.182 SPALTENNUMMER([Bereich])

SPALTENNUMMER([Bereich])

(Englisch: COLUMNNUMBER)

Diese Funktion gibt die Nummer der Spalte des angegebenen Bereichs (obere linke Ecke) oder der aktuellen Zelle an, falls kein Bereich eingegeben wurde. Die Zelle A1 hat dabei die Spaltennummer 1.

Diese Funktion kann vor allem mit Makros oder aber etwa mit der Funktion ZELLEABS gut benutzt werden. Ein Beispiel dazu finden Sie auch im Abschnitt <Namen-Namen festlegen>

Beispiele:

SPALTENNUMMER(F3) ergibt 6

SPALTENNUMMER(F3:H7) ergibt 6

SPALTENNUMMER() gibt die Spaltennummer der aktuellen Zelle zurück.

Verwandte Funktionen:

BLOCKBREITE, BLOCKHÖHE, ZEILENNUMMER, ZELLEABS, BEREICHABS

1.183 SVERWEIS(Wert;Bereich;Offset[:Exakt])

SVERWEIS(Wert;Bereich;Offset[:Exakt])

(Englisch: VLOOKUP)

Vertikales Suchen innerhalb von Bereich nach Wert: Diese Funktion sucht vertikal in der ersten Spalte von Bereich nach der ersten Zelle, deren Inhalt größer oder gleich Wert ist. Wird eine solche Zelle gefunden, gibt VLOOKUP den Wert der Zelle zurück, die sich Offset-Spalten rechts der Fundstelle befinden (ist Offset negativ, so entsprechend links der Stelle).

Sollte die Suche nicht erfolgreich sein, so wird #WERT zurückgegeben.

Dieser Befehl kann gut dazu benutzt werden, um einen bestimmten Wert in Abhängigkeit von einer Zelle aus einer Tabelle auszulesen.

Wert: Kann ein beliebiger Wert (Zahl, Text, Datum...) sein, nach dem gesucht wird.

Bei Text muß der Eintrag genau (bis auf Groß- und Kleinschreibung übereinstimmen). Bei allen anderen Werten wird (normalerweise) nach dem ersten größeren (oder gleichen) Wert gesucht.

Bereich: Der Bereich, innerhalb die Suche stattfindet. (Die Tabelle des Bereiches, d.h. die erste Spalte, sollte normalerweise sortiert sein!)

Offset: Bestimmt, wieviel Spalten rechts bzw. links der Wert in der gefundenen Zelle ausgelesen werden soll. (1=eins rechts, -1=eins links)

Exakt: Kann (und wird meist) ausgelassen. Wird der Parameter angegeben und ist er ungleich Null (etwa WAHR oder 1), so wird die "exakte" Suche ausgewählt, d.h. es wird nicht nach dem ersten größeren Wert, sondern nach genau dem Wert gesucht (Die Tabelle muß in diesem Fall nicht mehr sortiert sein!)

Hinweis: Es kann auch der Parameter Offset weggelassen werden, falls auch Exakt nicht angegeben wird. Offset wird in diesem Fall als 1 angenommen.

Beispiele:

SVERWEIS(10;A1:F5;1)

Verwandte Funktionen:

VERWEIS, WVERWEIS

1.184 TABELLENNAME()

TABELLENNAME()

(Englisch: SHEETNAME)

Gibt den Namen der Tabelle, in der sich die Formel befindet bzw. den aktuellen Namen im Makromodus zurück. Dieser Befehl kann im Zusammenhang mit dem Makrobefehl WÄHLETABELLE sinnvoll eingesetzt werden.

Hinweis: In der Version 4 wurde das Arbeitsmappenkonzept eingeführt, entsprechend wurden Funktionen hinzugefügt, mit denen der Mappen- bzw. der Blattname abgefragt werden kann. Um TABELLENNAME abwärtskompatibel zu gestalten, gibt er "Mappenname@Blattname" zurück. Dies kann, wie der Tabellename der Version3, mit AT dazu eingesetzt werden, um die Tabelle auszuwählen. Sie dazu auch bei AT.

Beispiele:

TABELLENNAME() ergibt (bis Version 3) in der Standardtabelle "Tabelle1" den Text "Tabelle1", ab der Version 4 wird "Tabelle1@Blatt1" retourniert.

Verwandte Funktionen:

BLATTNAME, MAPPENNAME

1.185 VERWEIS(Wert;Bereich[;Exakt])

VERWEIS(Wert;Bereich[;Exakt])

(Englisch: LOOKUP)

VERWEIS ist eine Abkürzung für WVERWEIS oder SVERWEIS (abhängig von Bereich) mit einem Offset von 1. Einzelheiten finden Sie bei den entsprechenden beiden Befehlen.

Normalerweise entspricht diese Funktion WVERWEIS(Wert;Bereich;1;Exakt). Ist der Bereich jedoch nur eine Spalte breit, so wird statt dessen SVERWEIS(Wert;Bereich;1;Exakt) berechnet.

Verwandte Funktionen:

WVERWEIS, SVERWEIS

1.186 WVERWEIS(Wert;Bereich;Offset[;Exakt])

WVERWEIS(Wert;Bereich;Offset[;Exakt])

(Englisch: HLOOKUP)

Horizontales Suchen innerhalb von Bereich nach Wert: Diese Funktion sucht horizontal in der ersten Reihe von Bereich nach der ersten Zelle, deren Inhalt größer oder gleich Wert ist. Wird eine solche Zelle gefunden, gibt HLOOKUP den Wert der Zelle zurück, die sich Offset-Zeilen unter der Fundstelle befinden (ist Offset negativ, so entsprechend über der Stelle)

Sollte die Suche nicht erfolgreich sein, so wird #WERT zurückgegeben.

Dieser Befehl kann gut dazu benutzt werden, um einen bestimmten Wert in Abhängigkeit von einer Zelle aus einer Tabelle auszulesen.

Wert: Kann ein beliebiger Wert (Zahl, Text, Datum...) sein, nach dem gesucht wird.

Bei Text muß der Eintrag genau (bis auf Groß- und Kleinschreibung übereinstimmen). Bei allen anderen Werten wird (normalerweise) nach dem ersten größeren (oder gleichen) Wert gesucht.

Bereich: Der Bereich, innerhalb die Suche stattfindet. (Die Tabelle des Bereiches, d.h. die erste Zeile, sollte normalerweise sortiert sein!)

Offset: Bestimmt, wieviel Zeilen tiefer bzw. höher der Wert in der gefundenen Zelle ausgelesen werden soll. (1=eins tiefer, -1=eins hoch)

Exakt: Kann (und wird meist) ausgelassen. Wird der Parameter angegeben und ist er ungleich Null (etwa WAHR oder 1), so wird die "exakte" Suche ausgewählt, d.h. es wird nicht nach dem ersten größeren Wert, sondern nach GENAU dem Wert gesucht (Die Tabelle muß in diesem Fall nicht mehr sortiert sein!)

Hinweis: Es kann auch der Parameter Offset weggelassen werden, falls auch Exakt nicht angegeben wird. Offset wird in diesem Fall als 1 angenommen.

Beispiele:

WVERWEIS(10;A1:F5;1)

HLOOKUP(10;A1:F5)

HLOOKUP(10;A1:F5;1;0)

habe alle die gleiche Funktion.

Etwa: Kosten pro Stück bei Abnahme von n Stück:

A B C D E

1: 5 10 50 100

2: 15 12 10 8

(Kosten bei Abnahme von max. n Stück)

Befindet sich in A3 dann die Stückzahl, so kann mit folgender Formel der Gesamtpreis berechnet werden: (Ergebnis für 3: 45, für 6: 60, für 10: 120)

=A3*HLOOKUP(A3;A1:E1;1)

Verwandte Funktionen:

SVERWEIS, VERWEIS

1.187 ZEILENNUMMER([Bereich])

ZEILENNUMMER([Bereich])

(Englisch: ROWNUMBER)

Diese Funktion gibt die Nummer der Zeile des angegebenen Bereichs (obere linke Ecke) oder der aktuellen Zelle an, falls kein Bereich eingegeben wurde. Die Zelle A1 hat dabei die Zeilennummer 1

Diese Funktion kann vor allem mit Makros oder aber etwa mit der Funktion ZELLEABS gut benutzt werden.

Beispiele:

ZEILENNUMMER(F3) ergibt 3

ZEILENNUMMER(F3:H7) ergibt 3

ZEILENNUMMER() gibt die Zeilennummer der aktuellen Zelle zurück.

Verwandte Funktionen:

BLOCKBREITE, BLOCKHÖHE, SPALTENNUMMER, ZELLEABS, BEREICHABS

1.188 ZELLE(Zeile;Spalte)

ZELLE(Zeile;Spalte)

(Englisch: CELL)

Die Funktion ZELLE liest den Wert aus der mit Spalte und Zeile relativ angegebenen Zelle aus und gibt ihn zurück.

Zeile und Spalte können beliebige ganze Zahlen ab 1 sein. 0 bedeutet die aktuelle Zeile/Spalte, negative Zahlen bestimmen, wie weit höher bzw. links die Zelle liegen, soll und positive geben die Richtung nach rechts unten an.

Bei der Bearbeitung von Makrobefehlen (also den Parametern dieser Befehle) gibt diese Funktion die Zelle relativ zur aktuellen Cursorposition des aktiven Fensters zurück. Hier bedeutet dann ZELLE(0;0) die Zelle, in der sich der Cursor befindet. (Ausnahme: Beim Parameter zum Makrobefehl SPRINGE und , siehe dort!)

Hinweis: Bitte beachten Sie, daß zuerst die Zeile und dann die Spalte kommt. Dies ist die Art, wie Tabellen normalerweise gelesen werden (entsprechend Matrizen!) - im Koordinatensystem würde dies jedoch Y,X bedeuten!

Beispiele:

ZELLE(-1;-1) ergibt den Wert, links über der aktuellen Zelle

Verwandte Funktionen:

ZELLEABS, INDIREKT, BEREICHABS

1.189 ZELLEABS(Zeile;Spalte)

ZELLEABS(Zeile;Spalte)

(Englisch: CELLABS)

Die Funktion ZELLEABS liest den Wert aus der mit Spalte und Zeile angegebenen Zelle aus und gibt ihn zurück.

Zeile und Spalte können beliebige positive Zahlen ab 1 sein (bei kleineren Werten erscheint #WERT)

Hinweis: Bitte beachten Sie, daß zuerst die Zeile und dann die Spalte kommt. Dies ist die Art, wie Tabellen normalerweise gelesen werden (entsprechend Matrizen!) - im Koordinatensystem würde dies jedoch Y,X bedeuten!

Tip: Diese Funktion kann gut in Makros benutzt werden, wenn man Zellen in Schleifen durchgehen möchte - siehe dazu das Beispiel bei .

Tip: Soll ein Block angegeben werden, so kann auch die Funktion ZELLEABS benutzt werden, etwa: ZELLEABS(1;3):ZELLEABS(7;5) einfacher ist jedoch BEREICHABS.

Beispiele:

ZELLEABS(1;1) ergibt den Wert, der in der Zelle A1 steht.

CELLABS(3;2) ergibt den Wert der Zelle B3

Verwandte Funktionen:

ZELLE, INDIREKT, BEREICHABS

1.190 Finanzfunktionen

Finanzfunktionen

ENDKAPITAL(Kapital;Zinssatz;Zeitraum[;Periode])

LAUFZEIT(Kapital;Endwert;Zinssatz[;Periode])

RATENENDKAPITAL(Ratenhöhe;Zinssatz;Zeitraum[;Periode])

RATENHÖHE(Endwert;Zinssatz;Zeitraum[;Periode])

RATENLAUFZEIT(Entwert;Ratenhöhe;Zinssatz[;Periode])

STARTKAPITAL(Endwert;Zinssatz;Zeitraum[;Periode])

ZINSSATZ(Kapital;Endwert;Zeitraum[;Periode])

TurboCalc verfügt auch über eine ganze Reihe von finanzmathematischen Funktionen. Diese benutzen als Parameter meist folgende Angaben:

Kapital: Ein fester Betrag, der als Grundlage für eine Berechnung dient. (Auch Start- bzw. Endkapital genannt.)

Ratenhöhe: Betrag, der monatlich... (siehe Periode) bezahlt wird.

Zinssatz: Zinssatz (also etwa 7.5% oder 0.075, jedoch nicht 7.5!), der eine Verzinsung beschreibt.

Zeitraum: Gibt den Zeitraum der Verzinsung an (also meist: Jahre). 1 entspricht in diesem Fall einem Jahr; 1/12 einem Monat, 1/360 einem (Bank-)Tag.

Periode: Bestimmt, wie oft pro "Jahr" die Verzinsung bzw. Ratenzahlung stattfinden soll. (1 bedeutet jährlich, 2 halbjährlich, 4 quartalsweise, 12 monatlich...) Dieser Parameter ist immer der letzte der Funktion und kann auch ausgelassen werden, dann wird 1 (also jährlich) angenommen.

Beispiele zu diesen Begriffen finden Sie bei den einzelnen Funktionen.

1.191 ENDKAPITAL(Kapital;Zinssatz;Zeitraum[;Periode])

ENDKAPITAL(Kapital;Zinssatz;Zeitraum[;Periode])

Diese Funktion berechnet das Endkapital eines einmalig angelegten Betrages. Einzugeben sind Anfangskapital, Laufzeit und Zinssatz.

Die übliche Fragestellung, die hiermit gelöst werden kann, lautet: "Auf welchen Betrag wächst ein bestimmter Betrag, den ich heute bei meiner Bank anlege, in einer festgelegten Laufzeit ?"

1.Beispiel:

Bei der Geburt eines Kindes wird auf einem Sparbuch ein Betrag von DM 2500,- zu einem jährlichen Zinssatz von 3.5% angelegt. Auf welchen Betrag wächst das Sparguthaben bis zum 18. Lebensjahr des Kindes an ?

ENDKAPITAL(2500;0.035;18)

Ergebnis: Das Kapital ist nach 18 Jahren auf DM 4643,72 angewachsen.

2.Beispiel:

Ein Betrag von DM 10000,- soll für 18 Monate als Festgeld angelegt werden. Die Bank bietet einen Zinssatz von 7.5% bei halbjährlicher Verzinsung. Auf welchen Betrag wächst das Kapital an ?

ENDKAPITAL(10000;7.5%;1.5;2)

Ergebnis: Das Kapital ist nach 18 Monaten auf DM 11167,71 angewachsen.

1.192 LAUFZEIT(Kapital;Endwert;Zinssatz[;Periode])

LAUFZEIT(Kapital;Endwert;Zinssatz[;Periode])

Der Funktion "LAUFZEIT" berechnet, wie lange ein Betrag bei bekanntem Zinssatz angelegt werden muß, um einen gewünschten Endwert zu erreichen.

Beispiele:

Ihr Sohn möchte sich gern einen neuen Computer kaufen. Er hat DM 4000,- zusammengespart. Leider kostet das Gerät, das er sich wünscht DM 4500,-. Wie lange muß er das Geld auf seinem Sparbuch (Zinssatz 3%) anlegen, damit er den erforderlichen Betrag erreicht ?

LAUFZEIT(4000;4500;3%)

Ergebnis: Ihr Sohn müßte (knapp) 4 Jahre warten, bis der Endwert erreicht ist.

1.193 RATENENDKAPITAL(Ratenhöhe;Zinssatz;Zeitraum[;Periode])

RATENENDKAPITAL(Ratenhöhe;Zinssatz;Zeitraum[;Periode])

Mit dieser Funktion können Sie ermitteln, welches Kapital Sie erreichen, wenn Sie über eine bestimmte Laufzeit in regelmäßigen Intervallen einen festen Betrag einzahlen. Vorausgesetzt wird ein über die gesamte Laufzeit gleichbleibender Zinssatz.

Beispiele:

Ein Auszubildender spart während seiner dreijährigen Ausbildungszeit monatlich einen Betrag von DM 200,-. Die Bank verzinst ihm seine Zahlungsreihe mit 6%, nach 3 Jahren möchte er sich von dem angesparten Kapital einen Gebrauchtwagen kaufen. Wieviel Geld wird er nach 3 Jahren für seinen Wagen zur Verfügung haben ?

RATENENDKAPITAL(200;6%;3;12)

Ergebnis: Nach seiner Ausbildung kann er sich einen Gebrauchtwagen im Wert von DM 7867,22 kaufen.

1.194 RATENHÖHE(Endwert;Zinssatz;Zeitraum[;Periode])

RATENHÖHE(Endwert;Zinssatz;Zeitraum[;Periode])

Diese Funktion benötigen Sie, wenn Sie zu einem bekannten oder gewünschten Endwert einer Zahlungsreihe die Höhe der regelmäßigen Einzahlungen ermitteln wollen. Es wird davon ausgegangen, daß der Zinssatz während der Laufzeit konstant bleibt.

Beispiele:

Eine Familie möchte sich in 5 Jahren eine neue Wohnzimmereinrichtung im Wert von DM 8000 kaufen. Die Bank bietet ihnen eine Verzinsung der regelmäßigen Einzahlungen in Höhe von 6.5% an. Wie hoch ist die Sparrate bei monatlichen Einzahlungen ?

RATENHÖHE(8000;6.5%;5;12)

Ergebnis: Die Familie muß monatlich DM 113,20 auf das Sparkonto einzahlen.

1.195 RATENLAUFZEIT(Entwert;Ratenhöhe;Zinssatz[;Periode])

RATENLAUFZEIT(Entwert;Ratenhöhe;Zinssatz[;Periode])

Mit dieser Funktion können Sie ermitteln, wie lange Sie eine Zahlungsreihe aufrechterhalten müssen, um ein gewünschtes Endkapital zu erreichen. Zinssatz, Ratenhöhe und die Häufigkeit der Ratenzahlungen müssen bekannt sein.

Beispiele:

Der Traum Ihres Lebens ist ein sündhaft teurer Sportwagen im Wert von DM 80000. Da Sie das Geld nicht zur Verfügung haben beschließen Sie, monatlich einen festen Betrag von DM 500,- zu sparen, bis das Kapital inklusive der Zinsen von jährlich 6.5% den erforderlichen Betrag erreicht. Wie lange müssen Sie nun diese Zahlungsreihe aufrechterhalten ?

RATENLAUFZEIT(80000;500;6.5%;12)

Ergebnis: Sie müssen noch 10 Jahre sparen, bis Sie sich den Wagen leisten können (sofern die Wagen bis dahin nicht teurer geworden ist!).

1.196 STARTKAPITAL(Endwert;Zinssatz;Zeitraum[;Periode])

STARTKAPITAL(Endwert;Zinssatz;Zeitraum[;Periode])

Diese Formel gestattet es, zu einem bekannten oder gewünschten Endwert ein anzulegendes Startkapital zu berechnen. Der Zinssatz pro Jahr, die Laufzeit und die Häufigkeit der Verzinsungen müssen ebenfalls bekannt sein.

Sie benötigen diese Funktion zum Beispiel, wenn Sie in absehbarer Zeit eine größere Ausgabe tätigen werden. Es wäre unnötig, den gesamten erforderlichen Betrag bereitzuhalten. Legen Sie statt dessen nur den Betrag an, der inklusive der Zinsen den benötigten Endbetrag ergibt.

Endwert: Hierbei handelt es sich um das Kapital, das Sie am Ende der Laufzeit erreichen möchten.

Beispiele:

Sie wollen sich in 3 Jahren ein neues Auto kaufen. Sie wissen, daß der Wagen ziemlich genau DM 30000,- kosten wird. Welchen Betrag müßten Sie jetzt bei der Bank anlegen, um in 3 Jahren den erforderlichen Betrag zu erhalten, wenn Ihnen die Bank 7.5% Zinsen bietet ?

STARTKAPITAL(30000;7.5%;3)

Ergebnis: Sie müssen heute DM 24148,82 anlegen, um in 3 Jahren den erforderlichen Betrag zur Verfügung zu haben.

1.197 ZINSSATZ(Kapital;Endwert;Zeitraum[;Periode])

ZINSSATZ(Kapital;Endwert;Zeitraum[;Periode])

Diese Funktion berechnet den Zinssatz eines einmalig angelegten Betrages. Anfangs- und Endwert müssen bekannt sein, ebenso die Laufzeit und die Häufigkeit der Zinszahlungen.

Beispiele:

Sie haben vor 2 Jahren bei Ihrer Bank einen Betrag von DM 6000,- festgelegt. Am Ende der Laufzeit war der Betrag auf DM 6933,80 angewachsen. Da Sie den zugehörigen Vertrag verlegt haben, versuchen Sie nun selbst herauszufinden, wie hoch der vereinbarte Zinssatz war.

ZINSSATZ(6000;69338.80;2)

Ergebnis: Ihr angelegtes Kapital wurde mit 7,5% verzinst.

1.198 Sonstige Funktionen

Sonstige Funktionen

ANIMBILD()

DATEIVORHANDEN(Datei)

DEMOVERSION()

INFO(Nr)

LETZTERFEHLER()

OBJEKTINFO(Name;Nr)

OBJID(Text)

REVISION()

SETZExxx(Bedingung;Wert1;Wert2[;Bezug])

TCFKT(TCLib;Offset;Num1;Num2;Text)

VERSION()

ZELLINFO(Nr[;Zelle])

1.199 ANIMBILD()

ANIMBILD()

(Englisch: ANIMFRAME)

Diese Funktion gibt innerhalb einer Diagrammanimation die aktuelle Bildnummer zurück (das erste Bild ist 0). Ansonsten wird der Wert -1 zurückgeliefert.

Diese Funktion ist für den Einsatz im Animationsmakro bestimmt. Einzelheiten finden Sie bei "Animation" im Kapitel "Diagramme" sowie bei der Beschreibung des Menüpunktes <Animation-Definieren>

1.200 DATEIVORHANDEN(Datei)

DATEIVORHANDEN(Datei)

(Englisch: FILEEXISTS)

Diese Funktion prüft, ob die angegebene Datei vorhanden ist und gibt entsprechend WAHR (falls vorhanden) oder FALSCH zurück.

Datei ist ein Text mit dem Dateinamen.

Hinweis: Obwohl dies eine Funktion ist, sollte sie nur im Zusammenhang mit Makros benutzt werden (siehe etwa Beispiel). Ansonsten sollte zumindest "automatisches Neuberechnen" deaktiviert werden. Wird dies nämlich als normale Funktion in der Tabelle eingesetzt, so wird bei jedem Neuberechnen geprüft, ob die Datei vorhanden ist, was das Neuberechnen sehr verzögern kann.

Anmerkung: Zum Prüfen werden keine Requester mit "Bitte Diskette... einlegen" angezeigt (falls ein Laufwerk nicht vorhanden ist, etwa bei "XXX:tes") sondern in diesem Fall sofort FALSCH zurückgegeben.

Beispiele:

```
=SPRINGEWENN(NOT(DATEIVORHANDEN("test.tcd"));A20)
```

```
=LADEN("test.tcd")
```

Dieses Fragment prüft, ob die Datei "test.tcd" vorhanden ist. Falls ja, wird sie geladen, ansonsten an die Stelle A20 zur Fehlerbehandlung verzweigt.

1.201 DEMOVERSION()

DEMOVERSION()

Diese Funktion gibt WAHR zurück, falls die aktuelle TurboCalc-Version nur eine Demoversion ist. Ansonsten wird FALSCH zurückgegeben.

Dies kann dazu benutzt werden, in Makros die aktuelle TurboCalc-Umgebung zu prüfen.

1.202 INFO(Nr)

INFO(Nr)

Die Funktion gibt Informationen zu einigen interessanten Gebieten zurück.

Nr. Resultat

0 Anzahl geöffneter Tabellen

1 Anzahl geöffneter Tabellenfenster

2 Anzahl geöffneter und sichtbarer Tabellenfenster (die ausgeblendeten zählen hierbei nicht!)

3 freier Chip-Speicher

4 freier Fast-Speicher

5 freier Speicher gesamt.

1.203 LETZTERFEHLER()

LETZTERFEHLER()

(Englisch: LASTERROR)

Gibt die Fehlernummer des letzten Fehlers zurück. Vor allem innerhalb einer Fehlerbehandlungsroutine interessant. Näheres dazu beim Makrobefehl BEIFEHLER().

1.204 OBJEKTINFO(Name;Nr)

OBJEKTINFO(Name;Nr)

(Englisch: OBJECTINFO)

Die Funktion gibt Auskunft über bestimmte Parameter eines Objekts.

Name: Der Name des Objekts. Existiert das Objekt nicht, erscheint die Fehlermeldung "WERT".

Nr. Information

0 Name (da der Name angegeben werden muß, nicht sonderlich sinnvoll, sondern nur aus Vollständigkeit vorhanden)

1 Makro-Text

2 Makro-aktiv (WAHR/FALSCH)

3 Hintergrundfarbe (0=aus, 1=Farbe 0...)

4 Rahmen (0=aus, 1=normal, 2=3D, 3=3D-2)

5 aktiviert (WAHR/FALSCH)

6 X-Position

7 Y-Position

8 Breite

9 Höhe

1.205 OBJID(Text)

OBJID(Text)

Diese Funktion berechnet aus einem Text (der Länge 1 bis 4) die entsprechende Objekttyp-ID für den Befehl OBJEKT. Dies erlaubt eine bessere Objekttypauswahl

Text: Ein Text, aus dem die Objekttyp-ID berechnet wird, nur die ersten vier Zeichen sind relevant, Groß- und Kleinschreibung wird unterscheiden. (Für Insider: Die Funktion summiert die ASCII-Werte nacheinander auf, jeweils mit 256^3 , 256^2 bzw. 256 multipliziert)

Beispiel

OBJID("TEXT") - siehe auch OBJEKT-Makro.

1.206 REVISION()

REVISION()

Diese Funktion gibt die Revisionsnummer der aktuellen TurboCalc-Version zurück (also die Nummer nach dem Komma).

Dies kann dazu benutzt werden, in Makros die aktuelle TurboCalc-Umgebung zu prüfen.

1.207 SETZExxx(Bedingung;Wert1;Wert2[;Bezug])

SETZExxx(Bedingung;Wert1;Wert2[;Bezug])

Diese Funktionen erlauben es, die Formatierung der aktuellen oder einer anderen Zelle abhängig von der Bedingung zu setzen/ändern. So ist es sehr einfach möglich (ohne Makroprogrammierung), die Tabelle je nach eingegebenen Daten automatisch zu ändern.

Sie lauten genau:

Befehl Wirkung

SETZEAUSRICHTUNG ändert die aktuelle Ausrichtung (links-, rechtsbündig...)

SETZEFARBE ändert die Schriftfarbe

SETZEFORMAT ändert das Zahlenformat

SETZESTIL ändert den Schriftstil

(Bzw. auf Englisch: SETALIGNMENT, SETCOLOR, SETFORMAT und SETSTYLE)

Bedingung ist dabei ein Wahrheitswert. Ist er WAHR (bzw. ungleich 0), so wird die entsprechende Formatierung auf Wert1 gesetzt, ansonsten auf Wert2.

Wert1, Wert2 sind die Werte, auf die das entsprechende Stilmerkmal je nach Bedingung gesetzt wird:

Bei Ausrichtung: 0=Standard, 1=Links, 2=Rechts, 3=Zentriert

Bei Farbe: 0=Standardfarbe, 1=Farbe1... (siehe auch beim Makrobefehl FARBEN)

Bei Format: 0=Standard, 1... gibt die Zahlenformatnummer an, siehe bei Makrobefehl ZAHLENFORMAT

Bei Stil: 0=Normal, 1=Unterstrichen, 2=Fett, 4=Kursiv (auch mehrere durch Addition möglich) siehe Makrobefehl SCHRIFT.

Bezug: Gibt die Zelle an, für die die Formatierung bestimmt werden soll. Fehlt dieser Eintrag, so wird die aktuelle Zelle genommen.

Alle Funktionen geben den Wert 0 zurück, falls die Änderung durchgeführt werden konnte. Ansonsten (etwa falls die Zielzelle leer ist!) wird eine entsprechende Fehlermeldung zurückgegeben.

Tip: Da diese Funktion etwas ändert und der Rückgabeparameter eigentlich uninteressant ist, kann er wie folgt sehr einfach ignoriert werden: Addieren Sie einfach diese Formel zu dem ursprünglichen Inhalt dazu (da ja 0 zurückgegeben wird, ändert dies nichts). Siehe auch erstes Beispiel unten.

Beispiele:

A1+SETZEFARBE(A1>=0;5;6) zeigt in der aktuellen Zelle den Wert der Zelle A1 an, und zwar in Farbe 5 falls er positiv ist und sonst in Farbe 6 (Dies kann also gut benutzt werden, um negative Einkünfte oder anderes besonders zu markieren).

SETZEFARBE(A1>0;5;6;A1) wie oben, jedoch wird die Farbe der Zelle A1 jetzt direkt geändert und es wird in der aktuellen Zelle einfach 0 zurückgegeben.

A1+SETZEAUSRICHTUNG(A1>=0;2;1) druckt den Inhalt von A1 aus und zwar rechtsbündig, falls er positiv ist und sonst linksbündig.

SUMME(A1:A10)+SETZESTIL(SUMME(A1:A10)>2000;3;4) berechnet das Ergebnis der Zellen A1 bis A10 und druckt es fett und unterstrichen, falls es größer als 2000 ist und sonst in kursiv.

1.208 TCFKT(TCLib;Offset;Num1;Num2;Text)

TCFKT(TCLib;Offset;Num1;Num2;Text)

(Englisch: TCFUNCTION)

Hiermit können externe Funktionen eingebunden und aufgerufen werden. Dies ist eine einfache Möglichkeit, TurboCalc um weitere Funktionen zu erweitern. Dazu werden TurboCalc-Libraries benutzt. Einzelheiten dazu finden im entsprechenden Abschnitt des Kapitels "Dateien".

TCLib: Dies ist der Name der TurboCalc-Library (als Text)

Offset: Gibt die Funktionsnummer an (somit können in einer Bibliothek mehrere Funktionen vorhanden sein).

Num1, Num2, Text: Sind die möglichen Parameter für die Funktion (Num1 und Num2 sind Zahlenwerte, Text ist ein Text)

Die genaue Funktionsweise, die Parameterbelegung sowie den Rückgabewert 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.

Hinweis: Möchten Sie externe Makrobefehle aufrufen, so können Sie dazu TCMAKRO benutzen.

1.209 VERSION()

VERSION()

Diese Funktion gibt die Versionsnummer der aktuellen TurboCalc-Version zurück (also die Nummer vor dem Komma).

Dies kann dazu benutzt werden, in Makros die aktuelle TurboCalc-Umgebung zu prüfen.

1.210 ZELLINFO(Nr[;Zelle])

ZELLINFO(Nr[;Zelle])

(Englisch: CELLINFO)

Diese Funktion liefert gewünschte Informationen zu einer bestimmten Zelle.

Nr gibt den Bereich an, zu dem Informationen gewünscht werden.

Zelle bestimmt die Zelle, auf die sich die Informationen befinden. Kann ausgelassen werden, dann wird die Zelle verwandt, in der sich die Formel befindet.

Nr. Information

0 Inhalt der Zelle (*immer* als Text)

1 Formel (als Text)

2 Datentyp (0: leere Zelle, 1: Fließkommazahl, 2: Ganzzahl, 3: Datum, 4: Zeit, 5: Boolean, 6: Text, 7: Zellbezug 8: Fehler) (7 normalerweise nicht möglich!)

3 Zahlenformat

4 Textfarbe

5 Hintergrundfarbe

6 Muster (0-15)

7 Textstil (1: Unterstrichen, 2: Fett, 4: Kursiv - und Summen daraus)

8 Horizontale Ausrichtung (0: Std., 1: Links, 2: Mitte, 3: Rechts)

9 Vertikale Ausrichtung (0: Std., 1: Oben, 3: Mitte, 2: Unten)

10 Rahmen (je zwei Bit (0-3, aus-max) je Seite -> 0-255!)

11 Schutzmerkmale (1: geschützt, 2: Formel verbergen - Summe!)

12 Spaltenbreite (hierbei wird die Spalte genommen, in der sich die angegebene Zelle befindet)

13 Zeilenhöhe (hierbei wird die Zeile genommen, in der sich die angegebene Zelle befindet)

14 Zellnotiz ("" falls die Zelle keine Notiz enthält)

Beispiele siehe gleichnamige Beispieltabelle!

1.211 Inhaltsverzeichnis

Inhaltsverzeichnis

Haupt-Inhaltsverzeichnis.

Kompletter Index (über alle Dateien).

Datentypen

Operatoren

Mathematikfunktionen

ABRUNDEN (Zahl;Stellen)

ABS(Zahl)

ARCCOS(Zahl)

ARCSIN(Zahl)

ARCTAN(Zahl)

AUFRUNDEN(Zahl;Stellen)

BOGEN(Zahl)

COS(Zahl)

COSHYP(Zahl)

EXP(Zahl)

EUKLID1 (Zahl1;Zahl2)

EUKLID2 (Zahl1;Zahl2)

FAKULTÄT(Zahl)

GANZZAHL(Zahl)

GERADE(Zahl)

GGT(Zahl1;Zahl2)

GRENZE(X;a;b;W1;W2;W3)

HOCH(Basis;Exponent)

HOCH10(Exponent)

KEHRWERT(Zahl)

KGV(Zahl1;Zahl2)

KOMBINATIONEN(n;k)

KUM([Zahl1;]Zahl2)

KÜRZEN(Zahl)

LG(Zahl)

LN(Zahl)

LOG(Zahl)

LOG10(Zahl)

MINVERS(Zahl;Modulo)

NACHKOMMA(Zahl)

OBERGRENZE(Zahl;Schritt)

PI()

QUADRAT(Zahl)

REST(Zahl1;Zahl2)

RUNDEN(Zahl;Stellen)

SIN(Zahl)

SINHYP(Zahl)

TAN(Zahl)

TANHYP(Zahl)

UNGERADE(Zahl)

UNTERGRENZE(Zahl;Schritt)

VARIATIONEN(n;k)

VORZEICHEN(Zahl)

WINKEL(Zahl)

WURZEL(Zahl[;Basis])

ZUFALLSBEREICH(Min;Max)

ZUFALLSZAHN()

ZWEIFAKULTÄT(Zahl)

Wahrheitswert-Funktionen

ÄHNLICH(Text1;Text2)

ENTHÄLT(Text1;Text2)

FALSCH()

IDENTISCH(Text1;Text2)

ISTDATUM(Wert)

ISTFEHLER(Bezug)

ISTGERADE(Zahl)

ISTLEER(Bezug)

ISTTEXT(Wert)

ISTUNGERADE(Zahl)

ISTZAHN(Wert)

ISTZEIT(Wert)

MUSTERVERGLEICH(Text;Muster)

NICHT(Wert)

ODER(Wert1;Wert2;)

UND(Wert1;Wert2;)

VERGLEICH(Text1;Text2)

WAHR()

WENN(Bedingung; Wert1; Wert2)

XOR(Wert1;Wert2;)

Text-Funktionen

Plus: +

BEHALTEN(String;Liste)

BERECHNEN(Formel)

CODE(Text)

BIN(Zahltext)

CONV2INT(Zahltext;Basis)

CONV2TEXT(Zahl;Basis)

ERSETZEN(String;Text1[:Text2])

GLÄTTEN(Text)

GROSS(Text)

GROSS2(Text)

HEX(Zahltext)

INTEXT(String;Muster[:Pos])

KLEIN(Text)

KOMPRIMIEREN(String[:Liste])

LÄNGE(Text)

LINKS(Text;Anzahl)

MITTE(Text;Zahl1;Zahl2)

NACHBIN(Zahl)

NACHHEX(Zahl)

NACHOCT(Zahl)

OCT(Zahltext)

RECHTS(Text;Anzahl)

RÖMISCH(Zahl)

SÄUBERN(Text)

SCHIEBENL(Text)

SCHIEBENR(Text)

SPIEGELN(Text)

TEIL(Text;Zahl1;Zahl2)

TEXT(Data[:Format])

WERT(Text)

WIEDERHOLEN(Text; Anzahl)

WORT(Text;Num)

WÖRTER(Text)

ZEICHEN(asc-code)

Datum- und Zeit-Funktionen

Plus: +, Minus: -

DATUM(Jahr;Monat;Tag)

DATWERT(Text)

HEUTE()

JAHR(Datum)

JETZT()

MINUTE(Zeit)

MINUTEN(Zeit)

MONAT(Datum)

MONATE(Datum)

MONATTEXT(Datum[:Abkürzung])

SEKUNDE(Zeit)

SEKUNDEN(Zeit)

STUNDE(Zeit)

STUNDEN(Zeit)

TAG(Datum)

TAGTEXT(Datum[:Abkürzung])

WOCHENTAG(Datum)

ZEIT(Stunde;Minute;Sekunde)

ZEITWERT(Text)

Tabellenfunktionen

ANZAHL(Bereich[:])

ANZAHL2(Bereich[:])

ERSTER(Bereich;...)

LETZTER(Bereich;...)

MAX(Bereich[:])

MIN(Bereich[:])

MITTELWERT(Bereich[:])

PRODUKT(Bereich[:])

STABW(Bereich[:])

SUMME(Bereich[:])

VANZAHL(Bereich[:])

VANZAHL2(Bereich[:])

VARIANZ(Bereich[:])

VERSTER(Bereich;...)

VLETZTER(Bereich;...)

VMAX(Bereich[:])

VMIN(Bereich[:])

VMITTELWERT(Bereich[:])

VPRODUKT(Bereich[:])

VSTABW(Bereich[:])

VSUMME(Bereich[:])

VVARIANZ(Bereich[:])

VWÄHLEN(Index;Bereich;...)

WÄHLEN(Index;Bereich;...)

Datenbankfunktionen

DBANZAHL(Datenbank;Spalte;Kriterien)

DBANZAHL2(Datenbank;Spalte;Kriterien)

DBERSTER(Datenbank;Spalte;Kriterien)

DBLETZTER(Datenbank;Spalte;Kriterien)
DBMAX(Datenbank;Spalte;Kriterien)
DBMIN(Datenbank;Spalte;Kriterien)
DBMITTELWERT(Datenbank;Spalte;Kriterien)
DBPRODUKT(Datenbank;Spalte;Kriterien)
DBSTABW(Datenbank;Spalte;Kriterien)
DBSUMME(Datenbank;Spalte;Kriterien)
DBVARIANZ(Datenbank;Spalte;Kriterien)
DBWÄHLEN(Index;Datenbank;Spalte;Kriterien)

Zellfunktionen

#Bezug
@Tabelle;Bezug
AKTUELLERBEREICH()
AUSWAHL(Index; Wert1; Wert2; Wert3)
BEREICHABS(Zeile;Spalte;Höhe;Breite)
BLATTNAME()
BLOCKBREITE([Bereich])
BLOCKHÖHE([Bereich])
BLOCKX([Bereich])
BLOCKY([Bereich])
INDIREKT(Text)
MAPPENNAME()
SPALTENNUMMER([Bereich])
SVERWEIS(Wert;Bereich;Offset[;Exakt])
TABELLENNAME()
VERWEIS(Wert;Bereich[;Exakt])
WVERWEIS(Wert;Bereich;Offset[;Exakt])
ZEILENUMMER([Bereich])
ZELLE(Zeile;Spalte)
ZELLEABS(Zeile;Spalte)

Finanzfunktionen

ENDKAPITAL(Kapital;Zinssatz;Zeitraum[;Periode])
LAUFZEIT(Kapital;Endwert;Zinssatz[;Periode])
RATENENDKAPITAL(Ratenhöhe;Zinssatz;Zeitraum[;Periode])
RATENHÖHE(Endwert;Zinssatz;Zeitraum[;Periode])
RATENLAUFZEIT(Entwert;Ratenhöhe;Zinssatz[;Periode])
STARTKAPITAL(Endwert;Zinssatz;Zeitraum[;Periode])
ZINSSATZ(Kapital;Endwert;Zeitraum[;Periode])

Sonstige Funktionen

ANIMBILD()
DATEIVORHANDEN(Datei)
DEMOVERSION()
INFO(Nr)
LETZTERFEHLER()
OBJEKTINFO(Name;Nr)
OBJID(Text)
REVISION()
SETZExxx(Bedingung;Wert1;Wert2[;Bezug])
TCFKT(TCLib;Offset;Num1;Num2;Text)
VERSION()
ZELLINFO(Nr[;Zelle])

1.212 Index

Index

Haupt-Inhaltsverzeichnis.

Kompletter Index (über alle Dateien).

- Symbole -

#Bezug

@Tabelle;Bezug

- A -

ABRUNDEN (Zahl;Stellen)

ABS(Zahl)

ÄHNLICH(Text1;Text2)

AKTUELLERBEREICH()

AND

ANIMBILD()

ANIMFRAME

ANZAHL(Bereich[;])

ANZAHL2(Bereich[;])

ARCCOS(Zahl)

ARCSIN(Zahl)

ARCTAN(Zahl)

AUFRUNDEN(Zahl;Stellen)

AUSWAHL(Index; Wert1; Wert2; Wert3)

AVERAGE

- B -

BEHALTEN(String;Liste)

BERECHNEN(Formel)

BEREICHABS(Zeile;Spalte;Höhe;Breite)

BIN(Zahltext)

BLATTNAME()

BLOCKBREITE([Bereich])

BLOCKHÖHE([Bereich])

BLOCKX([Bereich])

BLOCKY([Bereich])

BOGEN(Zahl)

- C -

CEILING

CELL

CELLABS

CELLINFO

CHAR

CHOOSE

CLEAN

CODE(Text)

COLUMNNUMBER

COMB

COMPARE

COMPRESS

CONTAINS

CONV2INT(Zahltext;Basis)

CONV2TEXT(Zahl;Basis)

COS(Zahl)

COSHYP(Zahl)

COUNT

COUNT2

CUM

CURRENTRANGE

- D -

DATE

DATEIVORHANDEN(Datei)

Datenbankfunktionen

Datentypen

DATEVALUE

DATUM(Jahr;Monat;Tag)

Datum- und Zeit-Funktionen

DATWERT(Text)
DAVERAGE
DAY
DBANZAHL(Datenbank;Spalte;Kriterien)
DBANZAHL2(Datenbank;Spalte;Kriterien)
DBERSTER(Datenbank;Spalte;Kriterien)
DBLETZTER(Datenbank;Spalte;Kriterien)
DBMAX(Datenbank;Spalte;Kriterien)
DBMIN(Datenbank;Spalte;Kriterien)
DBMITTELWERT(Datenbank;Spalte;Kriterien)
DBPRODUKT(Datenbank;Spalte;Kriterien)
DBSTABW(Datenbank;Spalte;Kriterien)
DBSUMME(Datenbank;Spalte;Kriterien)
DBVARIANZ(Datenbank;Spalte;Kriterien)
DBWÄHLEN(Index;Datenbank;Spalte;Kriterien)
DCOUNT
DCOUNT2
DEGTORAD
DEMOVERSION()
DFIRST
DLAST
DMAX
DMIN
DPICK
DPRODUCT
DSTDEV
DSUM
DVAR
- E -
ENDKAPITAL(Kapital;Zinssatz;Zeitraum[:Periode])
ENTHÄLT(Text1;Text2)
ERSETZEN(String;Text1[:Text2])
ERSTER(Bereich,...)
EUCLID1
EUCLID2
EUKLID1 (Zahl1;Zahl2)
EUKLID2 (Zahl1;Zahl2)
EVAL
EVEN

EXACT

EXP(Zahl)

- F -

FAC

FAKULTÄT(Zahl)

FALSCH()

FALSE

FILEEXISTS

Finanzfunktionen

FIRST

FLOOR

FOLDERNAME

FRAC

- G -

GANZ

GANZZAHL(Zahl)

GCD

GERADE(Zahl)

GGT(Zahl1;Zahl2)

GLÄTTEN(Text)

GRENZE(X;a;b;W1;W2;W3)

GROSS(Text)

GROSS2(Text)

- H -

HEUTE()

HEX(Zahltext)

HLOOKUP

HOCH(Basis;Exponent)

HOCH10(Exponent)

HOUR

HOURS

- I -

IDENTISCH(Text1;Text2)

IF

INDIRECT

INDIREKT(Text)

INFO(Nr)

INSTRING

INT

INTEGER

INTEXT(String;Muster[;Pos])

ISDATE

ISEMPTY

ISERROR

ISEVEN

ISODD

ISSTRING

ISTDATUM(Wert)

ISTFEHLER(Bezug)

ISTGERADE(Zahl)

ISTIME

ISTLEER(Bezug)

ISTTEXT(Wert)

ISTUNGERADE(Zahl)

ISTZAHL(Wert)

ISTZEIT(Wert)

ISVALUE

- J -

JAHR(Datum)

JETZT()

- K -

KEEP

KEHRWERT(Zahl)

KGV(Zahl1;Zahl2)

KLEIN(Text)

KOMBINATIONEN(n;k)

KOMPRIMIEREN(String[;Liste])

KUM([Zahl1;]Zahl2)

KÜRZEN(Zahl)

- L -

LAST

LASTERROR

LAUFZEIT(Kapital;Endwert;Zinssatz[;Periode])

LCM

LEFT

LENGTH

LETZTER(Bereich;...)

LETZTERFEHLER()

LG(Zahl)

LIMIT

LINKS(Text;Anzahl)

LN(Zahl)

LOG(Zahl)

LOG10(Zahl)

LOOKUP

LOWER

LÄNGE(Text)

- M -

MAPPENNAME()

MATCHES

Mathematikfunktionen

MAX(Bereich[;])

MID

MIN(Bereich[;])

MINUTE(Zeit)

MINUTEN(Zeit)

MINUTES

MINVERS(Zahl;Modulo)

MINVERSE

MITTE(Text;Zahl1;Zahl2)

MITTELWERT(Bereich[;])

MOD

MONAT(Datum)

MONATE(Datum)

MONATTEXT(Datum[;Abkürzung])

MONTH

MONTHS

MONTHTEXT

MUSTERVERGLEICH(Text;Muster)

- N -

NACHBIN(Zahl)

NACHHEX(Zahl)

NACHKOMMA(Zahl)

NACHOCT(Zahl)

NICHT(Wert)

NOT

NOW

- O -

OBBERGRENZE(Zahl;Schritt)

OBJECTINFO

OBJEKTINFO(Name;Nr)

OBJID(Text)

OCT(Zahltext)

ODD

ODER(Wert1;Wert2;)

Operatoren

OR

- P -

PART

PERM

PI()

PICK

Plus: +

Plus: +, Minus: -

POW

POW10

PRODUCT

PRODUKT(Bereich[:])

- Q -

QUADRAT(Zahl)

- R -

RADTODEG

RANGEABS

RANGEHEIGHT

RANGEWIDTH

RANGEX

RANGEY

RATENENDKAPITAL(Ratenhöhe;Zinssatz;Zeitraum[:Periode])

RATENHÖHE(Endwert;Zinssatz;Zeitraum[:Periode])

RATENLAUFZEIT(Entwert;Ratenhöhe;Zinssatz[:Periode])

RECHTS(Text;Anzahl)

RECIPROCAL

REPEAT

REST(Zahl1;Zahl2)

REVERSE

REVISION()

RIGHT

RND

RNDRANGE

ROMAN

ROUND

ROUNDDOWN

ROUNDUP

ROWNUMBER

RUNDEN(Zahl;Stellen)

RÖMISCH(Zahl)

- S -

SCHIEBENL(Text)

SCHIEBENR(Text)

SECOND

SECONDS

SEKUNDE(Zeit)

SEKUNDEN(Zeit)

SETALIGNMENT

SETCOLOR

SETFORMAT

SETSTYLE

SETZExxx(Bedingung;Wert1;Wert2[;Bezug])

SHEETNAME

SHEETNAME2

SHIFTL

SHIFTR

SIGN

SIMILAR

SIN(Zahl)

SINHYP(Zahl)

Sonstige Funktionen

SPALTENNUMMER([Bereich])

SPIEGELN(Text)

SQR

SQRT

STABW(Bereich[;])

STARTKAPITAL(Endwert;Zinssatz;Zeitraum[;Periode])

STDEV

STUNDE(Zeit)

STUNDEN(Zeit)

SUM

SUMME(Bereich[;])

SVERWEIS(Wert;Bereich;Offset[;Exakt])

SÄUBERN(Text)

- T -

Tabellenfunktionen

TABELLENNAME()

TAG(Datum)

TAGTEXT(Datum[;Abkürzung])

TAN(Zahl)

TANHYP(Zahl)

TCFKT(TCLib;Offset;Num1;Num2;Text)

TCFUNCTION

TEIL(Text;Zahl1;Zahl2)

TEXT(Data[;Format])

Text-Funktionen

TIME

TIMEVALUE

TOBIN

TODAY

TOHEX

TOOCT

TRANSLATE

TRIM

TRUE

TRUNC

TWOFAC

- U -

UND(Wert1;Wert2;)

UNGERADE(Zahl)

UNTERGRENZE(Zahl;Schritt)

UPPER

UPPER2

- V -

VALUE

VANZAHL(Bereich[;])

VANZAHL2(Bereich[;])

VAR

VARIANZ(Bereich[;])

VARIATIONEN(n;k)

VERSTER(Bereich;...)

VLETZTER(Bereich;...)

VMAX(Bereich[;])

VMIN(Bereich[;])

VMITTELWERT(Bereich[;])

VPRODUKT(Bereich[;])

VSTABW(Bereich[;])

VSUMME(Bereich[;])

VVARIANZ(Bereich[;])

VWÄHLEN(Index;Bereich;...)

VERGLEICH(Text1;Text2)

VERSION()

VERWEIS(Wert;Bereich[;Exakt])

VLOOKUP

VORZEICHEN(Zahl)

- W -

WAHR()

Wahrheitswert-Funktionen

WEEKDAY

WEEKDAYTEXT

WENN(Bedingung; Wert1; Wert2)

WERT(Text)

WIEDERHOLEN(Text; Anzahl)

WINKEL(Zahl)

WOCHENTAG(Datum)

WORD

WORDS

WORT(Text;Num)

WURZEL(Zahl[;Basis])

WVERWEIS(Wert;Bereich;Offset[;Exakt])

WÄHLEN(Index;Bereich;...)

WÖRTER(Text)

- X -

XOR(Wert1;Wert2;)

- Y -

YEAR

- Z -

ZEICHEN(asc-code)

ZEILENNUMMER([Bereich])

ZEIT(Stunde;Minute;Sekunde)

ZEITWERT(Text)

ZELLE(Zeile;Spalte)

ZELLEABS(Zeile;Spalte)

Zellfunktionen

ZELLINFO(Nr[;Zelle])

ZINSSATZ(Kapital;Endwert;Zeitraum[;Periode])

ZUFALLSBEREICH(Min;Max)

ZUFALLSZAHN()

ZWEIFAKULTÄT(Zahl)