Kalkulation (Funktionen)

siehe auch: [Kalkulation \(Artikel\)](#)
[Kalkulation \(Adressen\)](#)
[Inhaltsverzeichnis](#)

ABS	Bestimmt den absoluten Wert einer Zahl $ABS(x)$
ACOS	Berechnet den Arcus Cosinus einer Zahl $ACOS(x)$ $-1 \leq x \leq 1$ Ergebnis: Winkel in Bogenmaß (0..PI)
ACOSH	Berechnet den Cosinus Hyperbolicus einer Zahl $ACOSH(x)$ $x \geq 1$
ADDRESS	Bestimmt die Zellenadresse als Text $ADDRESS(\text{Zeile}, \text{Spalte}, \text{Ref-Typ} [, a1])$ Ref-Typ: 1 Zeile absolut, Spalte absolut 2 Zeile absolut, Spalte relativ 3 Zeile relativ, Spalte absolut 4 Zeile relativ, Spalte relativ
AND	Bestimmt den Wahrheitswert einer Liste $AND(1;2;3;0)$ oder $AND(F6:F10)$

Werte müssen logische Werte sein.

ASIN	Berechnet den Arcus Sinus einer Zahl $\text{ACOS}(x)$ $-1 \leq x \leq 1$ Ergebnis: Winkel in Bogenmaß $(-\pi/2..PI/2)$
ASINH	Berechnet den Sinus Hyperbolicus einer Zahl $\text{ASINH}(x)$
ATAN	Berechnet den Arcus Tangens einer Zahl $\text{ATAN}(x)$ Ergebnis: Winkel in Bogenmaß $(-\pi/2..PI/2)$
ATAN2	Berechnet den Arcus Tangens, der sich aus den angegebenen Koordinaten (x, y) ergibt. Dieser Arcus Tangens ist der Winkel zwischen der x-Achse und der Linie, die durch den Koordinatenursprung (0, 0) und den Punkt verläuft, der die Koordinaten (x, y) hat. $\text{ATAN2}(x,y)$ Ergebnis: Winkel in Bogenmaß $(-\pi/2..PI/2)$
ATANH	Berechnet den Arcus Tangens Hyperbolicus. $\text{ATANH}(x)$ $-1 < x < 1$
AVERAGE	Berechnet den Durchschnitt eines Bereiches

AVERAGE(x1; x2; x3;...)

AVERAGE(Zelle-Von:Zelle-Bis)

CALL

Aufruf einer Funktion in einer DLL

CALL(Dateiname, Funktionsname, Datentyp,
Argumente)

Datentyp

- A Logisch (falsch=0, wahr=1)
Wert, short int
- B IEEE8-Byte Fließkommazahl
Wert, double
- C Null-begrenzte Zeichenkette (255 Zeichen)
Referenz, char *
- D Zeichenkette mit Längenbyte (255 Zeichen)
Referenz, unsigned char *
- E IEEE8-Byte Fließkommazahl
Referenz, double *
- F Null-begrenzte Zeichenkette (255 Zeichen)
Referenz, char *
- G Zeichenkette mit Längenbyte (255 Zeichen)
Referenz, unsigned char *
- H vorzeichenlose ganze Zahl (2 Byte)
Wert, unsigned short int
- I vorzeichenbehaftete ganze Zahl (2 Byte)
Wert, short int
- J vorzeichenbehaftete ganze Zahl (4 Byte)
Wert, long int
- L Logisch (falsch=0, wahr=1)
Referenz, short int

M vorzeichenbehaftete ganze Zahl (2 Byte)
Referenz, short int

N vorzeichenbehaftete ganze Zahl (4 Byte)
Referenz, long int

CEILING Rundet einen Wert auf die festgelegte Position
`CEILING(Nummer, Position)`

CHAR gibt ein Zeichen entsprechend dem ANSI-Code zurück
`CHAR(nummer)`
 $0 \leq \text{nummer} \leq 255$

CHOOSE liefert den Wert aus einer Liste, der durch *index* bestimmt wird
`CHOOSE(index;1;2;3)`
index ist die Position in der Liste
1;2;3 stellt die Liste dar (können auch Zellen oder Strings sein)

CLEAN entfernt alle nichtdruckbaren Zeichen aus einem Text
`CLEAN(text)`
text ist eine Arbeitsblattinformation

CODE liefert Zahl des ersten Buchstabens einer Zeichenkette entsprechend ANSI-Code
`CODE(text)`

COLUMN	<p><i>text</i> ist eine Zeichenkette</p> <p>liefert die Spaltennummer</p> <p>COLUMN(x)</p> <p>x Verweis auf Spalte</p>
COLUMNS	<p>liefert die Anzahl der Spalten zwischen zwei vorgegebenen Spalten</p> <p>COLUMNS(bereich)</p> <p><i>bereich</i> ist die Anfangs- und Endspalte</p>
COS	<p>liefert den Cosinus einer Zahl</p> <p>COS(x)</p>
COSH	<p>liefert den Cosinus Hyperbolicus einer Zahl</p> <p>COSH(x)</p>
COUNT	<p>liefert die Anzahl der Elemente einer Liste</p> <p>COUNT(bereich)</p> <p><i>bereich</i> ist eine Liste von Einträgen</p>
COUNTA	<p>liefert die Anzahl der nichtleeren Elemente einer Liste</p> <p>COUNTA(bereich)</p> <p><i>bereich</i> ist eine Liste von Einträgen</p>
DATE	<p>liefert eine fortlaufende Nummer des angegebenen</p>

Datums

DATE(jahr; monat; tag)

jahr ist eine Jahreszahl zwischen 1900 und 2078. Ist das Jahr im Bereich zwischen 1920 und 2019 kann die Jahrhundertangabe entfallen.

monat ist eine Nummer die den Monat definiert.

tag beschreibt den Tag des Monats als Zahlenwert.

DATEVALUE

liefert eine fortlaufende Nummer des Datums. Die Datumsangabe steht in einer Zeichenkette.

DATEVALUE(text)

text ist das Datum im Bereich vom 1.1.1900 bis 31.12.2078. Für das aktuelle Jahr kann die Eingabe der Jahreszahl unterbleiben.

DAY

liefert die Nummer des Tages

DAY(nummer)

nummer ist eine Datumsangabe entweder als Nummer (1..31) oder als Text ("1.1.1999")

DB

Berechnet die Wertminderung eines Vermögens für eine definierte Periode unter Nutzung der deklinierten Ausgleichsmethode.

DB(kosten; rest; lebensdauer; periode [; monate])

kosten ist der Grundwert des Vermögens

rest ist der Restwert

lebensdauer ist die Anzahl der Perioden

periode ist der Zeitabschnitt für den die

Wertminderung gilt

monate ist die Anzahl der Monate im ersten Jahr

DDB

Liefert die degressive Doppelraten-Abschreibung eines Wirtschaftsgutes für eine bestimmte Periode.

DDB(*wert*; *restwert*; *nutzungsdauer*; *periode*; *faktor*)

wert sind die Anschaffungskosten eines Wirtschaftsgutes

restwert ist der Restwert am Ende der Nutzungsdauer

nutzungsdauer ist die Anzahl der Perioden, über die das Wirtschaftsgut abgeschrieben wird.

periode ist die Periode, deren Abschreibungsbetrag Sie berechnen möchten. Für das Argument *periode* muß dieselbe Zeiteinheit wie für die Nutzungsdauer verwendet werden.

faktor bestimmt die Rate, um die der jeweilige Buchwert verringert wird. Fehlt das Argument *faktor*, wird 2 vorausgesetzt (die degressive Doppelraten-Abschreibung).

Alle fünf Argumente müssen positive Zahlen sein.

Bei der Methode der degressiven Doppelraten-Abschreibung wird die Abschreibung entsprechend einer "beschleunigten" Rate berechnet. Die Abschreibung ist in der ersten Periode am höchsten und nimmt in den folgenden Perioden ab. DDB berechnet die Abschreibung einer Periode gemäß der folgenden Formel:

(Anschaffungskosten - Summe der Abschreibungen früherer Perioden) * Faktor / Nutzungsdauer

Für den Fall, daß Sie keine degressive Doppelraten-

Abschreibung wünschen, müssen Sie den Faktor ändern.

DOLLAR

Diese Funktion wandelt eine Zahl in einen Text um. Es wird das aktuelle Währungsformat verwendet und die Dezimalstellen entsprechend der angegebenen Stelle gerundet. Verwendetes Format: *###0,00 DM* oder *-###0,00 DM*.

DOLLAR(zahl; dezimalstellen)

zahl ist eine Zahl, ein Bezug auf eine Zelle, die eine Zahl enthält, oder eine Formel, deren Ergebnis eine Zahl ist.

dezimalstellen ist die Anzahl der Ziffern rechts vom Dezimalkomma. Ist *dezimalstellen* negativ, erfolgt das Runden von Zahl links vom Dezimalkomma. Wird das Argument Dezimalstellen nicht eingegeben, wird von 2 Dezimalstellen ausgegangen.

ERROR.TYPE

gibt eine Fehlernummer zurück

ERROR.TYPE(referenz)

referenz ist eine Arbeitsblattzelle

Nummer	Fehlertext
1	#NULL!
2	#DIV/0!
3	#VALUE!
4	#REF!
5	#NAME?
6	#NUM!
7	#N/A
#N/A	alle anderen Zellen

EVEN	<p>rundet den Wert auf die nächsthöhere ganze Zahl.</p> <p><code>EVEN(x)</code></p> <p><i>x</i> ist ein Zahlenwert.</p>
EXACT	<p>vergleicht zwei Zeichenketten auf Gleichheit</p> <p><code>EXACT(z1; z2)</code></p> <p>Sind die Zeichenketten <i>z1</i> und <i>z2</i> gleich ergibt dies True ansonsten False.</p>
EXP	<p>berechnet die Funktion e hoch x</p> <p><code>EXP(x)</code></p> <p><i>x</i> ist der Exponent</p> <p>$e=2,71828182845904$</p>
FACT	<p>ermittelt die Fakultät einer Zahl</p> <p><code>FACT(x)</code></p> <p><i>x</i> ist eine Zahl. Vor der Berechnung wird die Zahl zu einer ganzen Zahl gemacht.</p>
FALSE	<p>gibt den logischen Wert False (Falsch) zurück</p> <p><code>FALSE()</code></p>
FIND	<p>sucht in einer Zeichenkette nach einer anderen Zeichenkette</p> <p><code>FIND(such-text; text[; start-pos])</code></p> <p><i>such-text</i> ist zu suchender Text</p> <p><i>text</i> ist der Text in dem gesucht wird</p>

start-pos ist die Position im Text, ab der gesucht wird.

FIXED rundet den angegebenen Wert entsprechend *genauigkeit* und gibt Ergebnis als Text zurück

FIXED(wert [;*genauigkeit*][,*trennzeichen*])

wert ist der Zahlenwert

genauigkeit ist die Rundungsposition

trennzeichen legt fest ob ein tausender Trennzeichen dargestellt wird oder nicht

FLOOR rundet Wert auf das größte Vielfache von *schritt* ab

FLOOR(wert, *schritt*)

FV liefert ausgehend von regelmäßigen, konstanten Zahlungen und einem konstanten Zinssatz den zukünftigen Wert (Endwert) einer Investition (Kapitalanlage).

FV(*zins*; *zzr*; *rmz*; *bw*; *f*)

zins ist der Zinssatz pro Periode (Zahlungszeitraum).

zzr gibt an, über wie viele Perioden die jeweilige Rente gezahlt wird. (*zzr* = Anzahl der Zahlungszeiträume)

rmz ist der konstante Betrag (Annuität), der in jeder Periode gezahlt wird.

Üblicherweise umfaßt *rmz* das Kapital und die Zinsen, nicht jedoch sonstige Gebühren oder Steuern. (*rmz* = Regelmäßige Zahlung)

bw ist der Barwert oder der heutige Gesamtwert einer

Reihe zukünftiger Zahlungen. Fehlt das Argument *bw*, wird es als 0 angenommen. (*bw* = Barwert)

f kann den Wert 0 oder 1 annehmen und gibt an, wann Zahlungen fällig sind. Fehlt das Argument *f*, wird es als 0 angenommen. (*f* = Fälligkeit)

0 Am Ende einer Periode

1 Am Anfang einer Periode

HLOOKUP

Prüft, ob die oberste Zeile einer Matrix einen bestimmten Wert enthält, und liefert dann den Wert, den die angegebene Zeile enthält.

HLOOKUP können Sie immer dann einsetzen, wenn Ihre Vergleichswerte in der obersten Zeile einer Datentabelle stehen und Sie einen Wert lesen möchten, der sich eine bestimmte Anzahl an Zeilen unterhalb der Kopfzeile der Datentabelle befindet.

HLOOKUP(suchkriterium; matrix; zeilenindex; bereichsverweis)

suchkriterium ist der Wert, nach dem in der ersten Zeile der Matrix gesucht wird. *suchkriterium* kann ein Wert, ein Bezug oder eine Zeichenfolge in Anführungszeichen sein

matrix ist eine Informationstabelle, in der Daten gesucht werden. Geben Sie einen Bezug auf einen Bereich oder den Namen eines Bereichs an.

Die Werte, die die erste Zeile von *matrix* bilden, können Zeichenfolgen (Texte), Zahlen oder Wahrheitswerte sein.

Ist *bereichsverweis* gleich TRUE, müssen die in der ersten Zeile von *matrix* stehenden Werte in aufsteigender Reihenfolge angeordnet sein: ... -2, -1, 0, 1, 2, ... , A-Z, FALSCH, WAHR. Andernfalls besteht die Möglichkeit, daß HLOOKUP den falschen

Wert liefert.

Es erfolgt keine Unterscheidung bezüglich Groß- und Kleinbuchstaben.

Sie können die jeweiligen Werte in eine aufsteigende Reihenfolge bringen, indem Sie die Werte aufsteigend sortieren.

zeilenindex ist die Nummer der Zeile in der Mehrfachoperationsmatrix, aus der der übereinstimmende Wert geliefert werden soll. Ist *zeilenindex* gleich 1, wird der Wert geliefert, der in der ersten Zeile von *matrix* steht; ist *zeilenindex* gleich 2, wird der Wert geliefert, der in der zweiten Zeile von *matrix* steht usw. Ist *zeilenindex* kleiner als 1, liefert HLOOKUP den Fehlerwert #VALUE!; ist *zeilenindex* größer als die Anzahl der zu *matrix* gehörenden Zeilen, liefert HLOOKUP den Fehlerwert #REF!.

bereichsverweis ist ein Wahrheitswert, der angibt, wie die zu *matrix* gehörenden Werte sortiert sind. Ist *bereichsverweis* gleich TRUE oder nicht angegeben, müssen die Werte, die die erste Zeile von *matrix* bilden, in aufsteigender Reihenfolge vorliegen. Ist *bereichsverweis* gleich FALSE, spielt es keine Rolle, in welcher Reihenfolge die zu *matrix* gehörenden Werte angeordnet sind.

HOUR gibt die Stunde im 24 Stundenformat zurück
 HOUR(x)
 x ist die Zeit in der Form hh:mm

IF testet eine Bedingung
 IF(bedingung; wert-wahr, wert-falsch)
 bedingung ist eine logischer Vergleich

wert-wahr wird angenommen wenn *bedingung* wahr
wert-falsch wird angenommen, wenn *bedingung*
falsch

INDEX

liefert einen Wert aus einem oder mehreren Bereichen

`INDEX(bereich[;zeile][;spalte][;bereichsnummer])`

bereich definiert einen oder mehrere Bereiche auf dem Arbeitsblatt. Sind mehrere Bereiche ausgewählt, so sind diese in Klammern einzuschließen

zeile legt die ausgewählte Zeile fest

spalte legt die ausgewählte Spalte fest

bereichsnummer legt bei mehreren Bereichen den zu verwendenden fest.

Beispiel: `bereich (A5:B7;F3:H9)`

bereichsnummer 1 entspricht Bereich A5:B7

bereichsnummer 2 entspricht Bereich F3:H9

INDIRECT

verweist indirekt auf eine Zelle und nimmt deren Inhalt an

`INDEX(text[;a1])`

text ist die Zelle, die die Adresse einer anderen Zelle beinhaltet. Diese Zelle verweist ebenfalls auf eine andere Zelle, die den Wert beinhaltet

INT

rundet den Wert zum nächstgelegenen Integerwert ab

`INT(wert)`

IPMT

Diese Funktion liefert ausgehend von regelmäßigen,

konstanten Zahlungen und einem konstanten Zinssatz den Zinsbetrag, der in der angegebenen Periode gezahlt wird.

$IPMT(zins; zr; z zr; bw; zw; f)$

$zins$ ist der Zinssatz pro Periode.

zr ist die Periode, für die Sie den Zinsbetrag berechnen möchten. zr kann nur Werte im Bereich von 1 bis $z zr$ annehmen. (zr = Zeitraum)

$z zr$ gibt an, über wie viele Perioden die jeweilige Rente gezahlt wird. ($z zr$ = Anzahl der Zahlungszeiträume)

bw ist der Barwert oder der heutige Gesamtwert einer Reihe zukünftiger Zahlungen. (bw = Barwert)

zw ist der zukünftige Wert (Endwert) oder der Kassenbestand, den Sie nach der letzten Zahlung erreicht haben möchten. Fehlt das Argument zw , wird es als 0 angenommen (beispielsweise ist der Endwert eines Darlehens gleich 0).

f kann den Wert 0 oder 1 annehmen und gibt an, wann Zahlungen fällig sind. Fehlt das Argument f , wird es als 0 angenommen. (f = Fälligkeit)

0 Am Ende einer Periode

1 Am Anfang einer Periode

IRR

Liefert den internen Zinsfuß einer Investition ohne Finanzierungskosten oder Reinvestitionsgewinne. Die in Werte angegebenen Zahlen entsprechen der zu der Investition gehörenden Zahlungsreihe. Diese Zahlungen müssen nicht gleich groß sein, wie dies bei Annuitätenzahlungen der Fall ist. Der interne Zinsfuß ist der Zinssatz, der für eine Investition erreicht wird, die aus Auszahlungen (negative Werte) und Einzahlungen (positive Werte) besteht, die in

regelmäßigen Abständen erfolgen.

IRR(werte; schätzwert)

werte ist eine Matrix von Zellen oder ein Bezug auf Zellen, in denen die Zahlen stehen, für die Sie den internen Zinsfuß berechnen möchten.

werte muß mindestens einen positiven und einen negativen Wert enthalten, damit ein interner Zinsfuß berechnet werden kann.

IRR geht davon aus, daß die Zahlungen in der Reihenfolge erfolgen, wie sie in *werte* angegeben sind. Sie sollten daher darauf achten, daß Sie die Ein- und Auszahlungen in der von Ihnen gewünschten Reihenfolge eingeben.

Enthält ein als Matrix oder Bezug angegebenes Argument Text, Wahrheitswerte oder leere Zellen, werden diese Werte ignoriert.

schätzwert ist eine Zahl, von der Sie annehmen, daß sie dem Ergebnis der Funktion nahekommt.

Die Tabellenkalkulation in Mein System verwendet zur Berechnung der Funktion IRR ein Iterationsverfahren. Beginnend mit *schätzwert* wird die Funktion IRR solange ausgeführt, bis das Ergebnis auf 0,00001 Prozent genau ist. Kann IRR innerhalb von 20 Durchgängen kein geeignetes Ergebnis erzielen, wird der Fehlerwert #NUM! ausgegeben.

In den meisten Fällen ist es für die Berechnung von IRR nicht erforderlich, einen Schätzwert anzugeben. Fehlt das Argument schätzwert, wird es als 0,1 (10 Prozent) angenommen.

Liefert IRR den Fehlerwert #NUM!, oder weicht das Ergebnis stark von dem von Ihnen erwarteten Ergebnis ab, sollten Sie die Berechnung mit einem

anderen Schätzwert wiederholen.

ISBLANK	ermittelt ob die adressierte Zelle leer ist ISBLANK (referenz) <i>referenz</i> definiert Zielquelle True = wahr; False = falsch
ISERR	Liefert TRUE wenn der Ausdruck ein Fehler ist (ausgenommen Fehler #N/A!). Im anderen Fall wird FALSE zurückgegeben. ISERR (ausdruck)
ISERROR	Liefert TRUE wenn der Ausdruck ein Fehler ist (#N/A!, #VALUE!, #REF!, #DIV/0!, #NUM!, #NAME?, #NULL!). Im anderen Fall wird FALSE zurückgegeben. ISERR (ausdruck)
ISLOGICAL	Liefert TRUE wenn der Ausdruck einen logischen Wert liefert. Im anderen Fall wird FALSE zurückgegeben. ISLOGICAL (ausdruck)
ISNA	Liefert TRUE wenn der Ausdruck Fehler #N/A! ist. Im anderen Fall wird FALSE zurückgegeben. ISNA (ausdruck)
ISNONTEXT	Liefert TRUE wenn der Ausdruck keinen Text

zurückgibt. Im anderen Fall wird FALSE zurückgegeben.

ISNONTEXT(ausdruck)

ISNUMBER Liefert TRUE wenn der Ausdruck eine Zahl zurückgibt. Im anderen Fall wird FALSE zurückgegeben.

ISNUMBER(ausdruck)

ISREF Liefert TRUE wenn der Ausdruck ein Bezug ist (z.B. B30). Im anderen Fall wird FALSE zurückgegeben.

ISREF(ausdruck)

ISTEXT Liefert TRUE wenn der Ausdruck einen Text ergibt. Im anderen Fall wird FALSE zurückgegeben.

ISTEXT(ausdruck)

LEFT Liefert den ersten Buchstaben eines Strings zurück.

Left(text[; anz])

text ist der String

anz ist die Anzahl der zurückgelieferten Buchstaben. Wird *anz* nicht angegeben wird der Wert 1 verwendet.

LEN Liefert die Länge des durch den Ausdruck definierten Strings zurück.

LEN(ausdruck)

LN	<p>Liefert den natürlichen Logarithmus einer Zahl. Natürliche Logarithmen haben die Konstante e (2,71828182845904) als Basis.</p> <p>LN(zahl)</p> <p><i>zahl</i> ist eine Zahl deren natürlichen Logarithmus berechnet werden soll.</p>
LOG	<p>Liefert den Logarithmus einer Zahl zur gewählten Basis.</p> <p>LOG(zahl[;basis])</p> <p><i>zahl</i> ist eine positive reelle Zahl deren Logarithmus berechnet werden soll.</p> <p><i>basis</i> ist die Basis des Logarithmus. Wird diese nicht angegeben, so wird als Basis die 10 verwendet.</p>
LOG10	<p>Liefert den Logarithmus einer Zahl zur Basis 10.</p> <p>LOG10(zahl)</p> <p><i>zahl</i> ist eine positive reelle Zahl deren Logarithmus berechnet werden soll.</p>
LOOKUP	<p>Die Matrixversion von LOOKUP arbeitet wie folgt: Sie sucht innerhalb der ersten Zeile oder Spalte einer Matrix nach dem jeweils angegeben Wert, springt nach unten bzw. rechts in die letzte Zelle und liefert den in dieser Zelle stehenden Wert. In der Regel empfiehlt es sich, anstelle von LOOKUP eine der Funktionen HLOOKUP oder VLOOKUP einzusetzen. Die hier erläuterte Version von LOOKUP steht nur aus Gründen der Kompatibilität mit anderen Tabellenkalkulationsprogrammen zur Verfügung.</p>

LOOKUP(suchkriterium; suchvektor)

suchkriterium ist ein Wert, nach dem LOOKUP in der Matrix sucht. *suchkriterium* kann eine Zahl, eine Zeichenfolge (ein Text), ein Wahrheitswert oder ein Name bzw. ein Bezug sein, der sich auf einen Wert bezieht.

Kann die Funktion LOOKUP keinen Wert finden, der mit dem jeweiligen *suchkriterium* übereinstimmt, verwendet sie den zu der Suchvektor gehörenden Wert, der bezogen auf das Suchkriterium der nächstkleinere ist.

Ist das Suchkriterium kleiner als der kleinste Wert der ersten Zeile oder Spalte (dies hängt von den Matrixdimensionen ab), liefert LOOKUP den Fehlerwert #NA.

suchvektor ist ein Zellbereich, der entweder Text, Zahlen oder Wahrheitswerte enthält, die Sie mit den Suchkriterien vergleichen möchten.

Ist der *suchvektor* quadratisch oder hat er mehr Spalten als Zeilen, gleicht LOOKUP die Elemente der ersten Zeile gegen das Suchkriterium ab.

Hat der Suchvektor mehr Zeilen als Spalten, durchsucht LOOKUP die erste Spalte.

HLOOKUP und VLOOKUP bieten die Möglichkeit, durch die Angabe eines Indexes in eine bestimmte Zeile oder Spalte zu springen. LOOKUP wählt dagegen immer den Wert, der in der letzten Zeile bzw. Spalte steht.

Wichtig! Die Werte müssen in aufsteigender Reihenfolge angeordnet sein: ..., -2, -1, 0, 1, 2, ..., A-Z, FALSE, TRUE, andernfalls liefert LOOKUP eventuell einen falschen Wert. Bei Zeichenfolgen (Texten) wird nicht nach Groß- und Kleinbuchstaben

unterschieden.

LOWER

Wandelt die Buchstaben in dem ausgewählten String in Kleinbuchstaben.

LOWER(text)

MATCH

Liefert die relative Position, die ein Element, das auf bestimmte Weise mit einem angegebenen Wert übereinstimmt, innerhalb einer Matrix einnimmt.

MATCH(suchkriterium; suchmatrix; vergleichstyp)

suchkriterium ist der Wert, der gesucht werden soll

suchmatrix ist der Bereich, in dem gesucht werden soll

vergleichstyp kann die zahlen -1, 0 und 1 annehmen

Ist *vergleichstyp* gleich 1, liefert MATCH den größten Wert, der kleiner gleich dem Suchkriterium ist. Die Elemente der *suchmatrix* müssen in aufsteigender Reihenfolge angeordnet sein: ...-2, -1, 0, 1, 2,...A-Z, FALSE, TRUE.

Ist *vergleichstyp* gleich 0, liefert MATCH den ersten Wert, der gleich dem Suchkriterium ist. Die Elemente der Suchmatrix dürfen in beliebiger Reihenfolge angeordnet sein.

Ist *vergleichstyp* gleich -1, liefert MATCH den kleinsten Wert, der größer gleich dem Suchkriterium ist. Die Elemente der Suchmatrix müssen in absteigender Reihenfolge angeordnet sein: TRUE, FALSE, Z-A,...2, 1, 0, -1, -2,... usw.

Fehlt *vergleichstyp*, wird es als 1 angenommen.

MAX	<p>Liefert den größten Wert einer Argumentenliste.</p> <p>MAX(liste)</p> <p>liste ist ein Bereich von 1 bis 30 Zahlen, deren größte bestimmt werden soll.</p>
MID	<p>Liefert einen Teilstring aus einem String</p> <p>MID(text;beginn,anz)</p> <p><i>text</i> ist der Gesamtstring</p> <p><i>beginn</i> ist die Anfangsposition des Teilstrings im Gesamtstring</p> <p><i>anz</i> ist die Anzahl der Buchstaben des Teilstrings</p>
MIN	<p>Liefert den kleinsten Wert einer Argumentenliste.</p> <p>MIN(liste)</p> <p><i>liste</i> ist ein Bereich von 1 bis 30 Zahlen, deren kleinste bestimmt werden soll.</p>
MINUTE	<p>gibt die Minute zurück</p> <p>MINUTE(x)</p> <p><i>x</i> ist die Zeit in der Form hh:mm als Text oder ein Zahlenwert</p>
MIRR	<p>Diese Funktion liefert den modifizierten internen Zinsfuß, der sich bei einer Reihe periodischer Zahlungen ergibt. MIRR berücksichtigt sowohl die Kosten der jeweiligen Investition als auch die Zinsen, die sich aus der Reinvestition des Geldes ergeben. (MIRR = Qualifizierter interner Kapitalverzinsungs-</p>

satz)

MIRR(werte; investition; reinvestition)

werte ist eine Matrix oder ein Bezug auf Zellen, die Zahlen enthalten. Diese Zahlen entsprechen einer Reihe von Auszahlungen (negative Werte) sowie Einzahlungen (positive Werte), die in gleichlangen Perioden erfolgen.

werte muß mindestens einen positiven und einen negativen Wert umfassen, damit der modifizierte interne Zinsfuß berechnet werden kann. Andernfalls liefert MIRR den Fehlerwert #DIV/0!.

Enthält ein als Matrix oder Bezug angegebenes Argument Text, Wahrheitswerte oder leere Zellen, werden diese Werte ignoriert. Zellen, die den Wert 0 enthalten, werden dagegen berücksichtigt.

investition ist der Zinssatz, den Sie für die gezahlten Gelder ansetzen.

reinvestition ist der Zinssatz, den Sie für reinvestierte Gelder erzielen.

MOD

Liefert den Rest der Division von Zahl durch Divisor.

MOD(zahl;divisor)

MONTH

gibt den Monat zurück

MONTH(x)

x ist das Datum in der Form TT.MM.JJ als Text oder ein Zahlenwert.

z.B. "6.Juli.1999"; 5,3

N Diese Funktion liefert die Zahl, die sich aus der Umwandlung des als Argument übergebenen Wertes ergibt.

N(wert)

wert ist der Wert, den Sie in eine Zahl umwandeln möchten. N wandelt Werte folgendermaßen um:

Zahl	Zahl
Datum	fortlaufende Zahl dieses Datums
TRUE	1
Sonstige	0

NA Liefert den Fehlerwert #NA. Der Fehlerwert #N/A besagt, daß "kein Wert verfügbar" ist. Verwenden Sie NA, um leere Zellen zu kennzeichnen. Indem Sie #N/A in Zellen eingeben, die keine Informationen enthalten, können Sie verhindern, daß leere Zellen unbeabsichtigt in Ihre Berechnungen einbezogen werden. (Wenn sich eine Formel auf eine Zelle bezieht, die den Wert #N/A enthält, liefert die Formel den Fehlerwert #NA.)

NA()

NOT Diese Funktion kehrt den Wert ihres Argumentes um. NOT können Sie immer dann einsetzen, wenn Sie sicherstellen möchten, daß ein Wert nicht mit einem bestimmten Wert übereinstimmt.

NOT(wahrheitswert)

wahrheitswert ist ein Wert oder Ausdruck, der einen der Werte TRUE oder FALSE annehmen kann. Ist *wahrheitswert* mit FALSE belegt, liefert NOT den

Wert TRUE; ist *wahrheitswert* mit TRUE belegt, liefert NOT den Wert FALSE.

NOW

Liefert die fortlaufende Zahl des aktuellen Datums und der aktuellen Uhrzeit.

JETZT()

NPER

Diese Funktion liefert die Anzahl der Zahlungsperioden für eine Investition, die auf periodischen, gleichbleibenden Zahlungen sowie einem konstanten Zinssatz basiert.

NPER(zins; rmz; bw; zw; f)

zins ist der Zinssatz pro Periode.

rmz ist der Betrag (die Annuität), der in jeder Periode gezahlt wird. Dieser Betrag kann sich während der Laufzeit nicht ändern.

Üblicherweise umfaßt *rmz* das Kapital und die Zinsen, nicht jedoch sonstige Gebühren oder Steuern. (*rmz* = Regelmäßige Zahlung)

bw ist der Barwert oder der heutige Gesamtwert einer Reihe zukünftiger Zahlungen.

zw ist der zukünftige Wert (Endwert) oder der Kassenbestand, den Sie nach der letzten Zahlung erreicht haben möchten. Fehlt das Argument *zw*, wird es als 0 angenommen (beispielsweise ist der Endwert eines Darlehens gleich 0).

f kann den Wert 0 oder 1 annehmen und gibt an, wann Zahlungen fällig sind.

0 Am Ende einer Periode

1 Am Anfang einer Periode

NPV

Liefert den Nettobarwert (Kapitalwert) einer Investition auf Basis eines Abzinsungsfaktors für eine Reihe periodischer Zahlungen. Der Nettobarwert einer Investition ist der gegenwärtige Wert einer Reihe zukünftiger Auszahlungen (negative Werte) und Einzahlungen (positive Werte).

$NPV(zins, wert1, wert2, \dots)$

zins ist der Abzinsungssatz für die Dauer einer Periode.

wert1; wert2;... sind 1 bis 29 Argumente, die den Auszahlungen und den Einzahlungen entsprechen.

wert1; wert2;... müssen als Zahlungsvorgänge in gleichbleibenden Zeitabständen erfolgen und sind jeweils am Ende einer Periode vorzunehmen.

NPV bestimmt anhand der Reihenfolge von *wert1; wert2;...* die Reihenfolge der Zahlungen. Sie müssen daher darauf achten, daß Sie die Auszahlungen und Einzahlungen in der richtigen Reihenfolge eingeben.

Es werden alle Argumente berücksichtigt, bei denen es sich um Zahlen, leere Zellen, Wahrheitswerte oder Zahlen in Textform handelt. Argumente, die Fehlerwerte oder Text enthalten, der sich nicht in eine Zahl umwandeln läßt, werden ignoriert.

Ist als Argument eine Matrix oder ein Bezug angegeben, werden nur die Elemente der Matrix oder des Bezuges berücksichtigt, die Zahlen enthalten. In der Matrix oder dem Bezug stehende leere Zellen, Wahrheitswerte, Texte oder Fehlerwerte werden ignoriert.

ODD

Diese Funktion rundet Zahl auf die nächste ungerade

ganze Zahl auf.

ODD(zahl)

zahl ist der Wert, den Sie runden möchten.

OFFSET

Liefert einen Bezug, der einen Bereich bestimmter Höhe und Breite angibt und gegenüber einem anderen Bezug um eine bestimmte Anzahl an Zeilen und Spalten versetzt ist.

OFFSET(*bezug*; *zeilen*; *spalten*; *höhe*; *breite*)

bezug ist der Bezug, der als Ausgangspunkt des Verschiebevorgangs dienen soll. Ist *bezug* eine Mehrfachmarkierung, liefert OFFSET den Fehlerwert #VALUE!.

zeilen ist die Anzahl der Zeilen, um die Sie die obere linke Eckzelle des Bereiches nach oben oder nach unten verschieben wollen. Ist das Argument *zeilen* beispielsweise gleich 5, so bedeutet dies, daß die obere linke Ecke des neuen Bezugs fünf Zeilen unterhalb von Bezug liegt. *zeilen* darf sowohl positive als auch negative Werte annehmen.

spalten ist die Anzahl der Spalten, um die Sie die obere linke Eckzelle des Bereiches nach links oder nach rechts verschieben wollen. Ist das Argument *spalten* beispielsweise gleich 5, so bedeutet dies, daß die obere linke Ecke des neuen Bezugs fünf Spalten rechts von Bezug liegt. *spalten* darf sowohl positive als auch negative Werte annehmen.

Wenn die Werte von *zeilen* und *spalten* zur Folge haben, daß der Bezug über den Rand der jeweiligen Tabelle hinaus versetzt wird, liefert OFFSET den Fehlerwert #REF!.

höhe ist die Höhe des neuen Bezuges in Zeilen. *höhe*

muß eine positive Zahl sein.

breite ist die Breite des neuen Bezuges in Spalten. breite muß eine positive Zahl sein.

Fehlt das Argument *höhe* oder *breite*, geht Mein System davon aus, daß der neue Bezug dieselbe Höhe oder Breite wie *bezug* hat.

OR

Liefert TRUE, wenn ein Argument TRUE ist; liefert FALSE, wenn alle Argumente FALSE sind.

OR(*wert1*; *wert2*; ...)

wert1; *wert2*;... sind 1 bis 30 Bedingungen, die überprüft werden sollen und jeweils entweder TRUE oder FALSE sind.

Geben Sie als Argumente nur Wahrheitswerte oder aber Matrizen oder Bezüge an, die Wahrheitswerte enthalten.

Enthält ein als Matrix oder Bezug angegebenes Argument Text, Zahlen oder leere Zellen, werden diese Werte ignoriert.

Enthält ein angegebener Bereich keine Wahrheitswerte, liefert OR den Fehlerwert #VALUE!.

PI

liefert den Wert der Zahl PI (3,14159265358979) mit einer Genauigkeit von 15 Stellen.

PMT

Diese Funktion liefert die Annuität eines Darlehens oder einer Investition, wobei ein konstanter Zinssatz vorausgesetzt wird.

PMT(*zins*; *zsr*; *bw*; *zw*; *f*)

zins ist der Zinssatz pro Periode.

zr gibt an, über wie viele Perioden die jeweilige Annuität (Rente) gezahlt wird. (zr = Anzahl der Zahlungszeiträume)

bw ist der Barwert: der Gesamtbetrag, den eine Reihe zukünftiger Zahlungen zum gegenwärtigen Zeitpunkt wert ist.

zw ist der zukünftige Wert (Endwert) oder der Kassenbestand, den Sie nach der letzten Zahlung erreicht haben möchten. Fehlt das Argument zw , wird es als 0 angenommen (beispielsweise ist der Endwert eines Darlehens gleich 0). (zw = Zukünftiger Wert)

f kann den Wert 0 oder 1 annehmen und gibt an, wann Zahlungen fällig sind.

0 Am Ende einer Periode

1 Am Anfang einer Periode

PPMT

Diese Funktion berechnet für die jeweils angegebene Periode, wie groß der Tilgungsanteil der für geliehenes Kapital fälligen Rückzahlungen ist. Für diese Rückzahlungen werden gleichbleibende Beträge und ein konstanter Zinssatz vorausgesetzt. (Kapitalrückzahlung)

$PPMT(zins; zr; zcr; bw; zw; f)$

$zins$ ist der Zinssatz pro Periode.

zr gibt die Periode an und muß in dem Bereich von 1 bis zcr liegen. (zr = Zahlungszeitraum)

zcr gibt an, über wie viele Perioden die jeweilige Annuität (Rente) gezahlt wird. (zcr = Anzahl der Zahlungszeiträume)

bw ist der Barwert: der Gesamtbetrag, den eine Reihe zukünftiger Zahlungen zum gegenwärtigen Zeitpunkt wert ist.

zw ist der zukünftige Wert (Endwert) oder der Kassenbestand, den Sie nach der letzten Zahlung erreicht haben möchten. Fehlt das Argument zw , wird es als 0 angenommen (beispielsweise ist der Endwert eines Darlehens gleich 0).

f kann den Wert 0 oder 1 annehmen und gibt an, wann die Zahlungen fällig sind.

0 Am Ende einer Periode

1 Am Anfang einer Periode

PRODUCT

Diese Funktion multipliziert alle als Argumente angegebenen Zahlen miteinander und liefert das Produkt.

PRODUCT(zahl1; zahl2; ...)

zahl1; zahl2;... sind 1 bis 30 Zahlen, die Sie miteinander multiplizieren möchten.

PROPER

Wandelt den ersten Buchstaben von Text sowie alle zu Text gehörenden Buchstaben, die unmittelbar hinter einem Zeichen stehen, das kein Buchstabe ist, in Großbuchstaben um. Wandelt alle anderen Buchstaben in Kleinbuchstaben um.

Proper(text)

text ist ein in Anführungszeichen eingeschlossener Text, eine Formel, die einen Text liefert, oder ein Bezug auf eine Zelle, die den Text enthält.

PV

Liefert den Barwert einer Investition. Der Barwert ist der Gesamtbetrag, den eine Reihe zukünftiger Zahlungen zum jetzigen Zeitpunkt wert ist. Wenn Sie beispielsweise einen Kredit aufnehmen, ist die Summe dieses Kredits für den Kreditgeber gleich dem

Barwert.

$PV(\text{zins}; \text{zrz}; \text{rmz}; \text{zw}; \text{f})$

zins ist der Zinssatz pro Periode. Wenn Sie beispielsweise für die Anschaffung eines Autos ein Darlehen mit einem jährlichen Zinssatz von 10% aufnehmen und dieses Darlehen in monatlichen Raten zurückzahlen, beträgt der monatliche Zinssatz $10\%/12$ oder 0,83%. Für den Zinssatz müßten Sie also einen der Werte $10\%/12$ oder 0,83% oder 0,0083 in die Formel eingeben.

zrz gibt an, über wie viele Perioden die jeweilige Annuität (Rente) gezahlt wird. (*zrz* = Anzahl der Zahlungszeiträume). Wenn Sie beispielsweise zur Finanzierung einer Maschine einen Kredit mit einer Laufzeit von 5 Jahren aufnehmen, den Sie in monatlichen Raten zurückzahlen, hat der Kredit eine Laufzeit von $5 \cdot 12$ (oder 60) Perioden. Für *zrz* müßten Sie also 60 angeben.

rmz ist der Betrag (die Annuität), der in jeder Periode gezahlt wird. Dieser Betrag kann sich während der Laufzeit nicht ändern. Üblicherweise umfaßt *rmz* das Kapital und die Zinsen, nicht jedoch sonstige Gebühren oder Steuern. (*rmz* = Regelmäßige Zahlung). Beispielsweise müssen Sie für einen Autokredit, der bei einem Zinssatz von 12% eine Laufzeit von fünf Jahren hat, monatlich 270,00 DM zurückzahlen. Für *rmz* müßten Sie also -270,00 angeben.

zw ist der zukünftige Wert (Endwert) oder der Kassenbestand, den Sie nach der letzten Zahlung erreicht haben möchten. Fehlt das Argument *zw*, wird es als 0 angenommen (beispielsweise ist der Endwert eines Darlehens gleich 0). Wenn Sie als Beispiel 100.000 DM ansparen möchten, um in 20 Jahren ein bestimmtes Projekt finanzieren zu können, ist der

zugehörige Endwert 100.000 DM. Mit einer vorsichtigen Schätzung des Zinssatzes können Sie nun ausrechnen, wieviel Sie jeden Monat sparen müssen.

f kann den Wert 0 oder 1 annehmen und gibt an, wann die Zahlungen fällig sind.

0 Am Ende einer Periode

1 Am Anfang einer Periode

RAND

Diese Funktion liefert eine gleichverteilte Zufallszahl, die größer gleich 0 und kleiner als 1 ist. Bei jeder Neuberechnung der jeweiligen Tabelle wird eine neue Zufallszahl ausgegeben.

`RAND()`

RATE

Liefert den Zinssatz einer Annuität pro Periode. RATE verwendet zur Berechnung eines Zinssatzes ein Iterationsverfahren. Es ist sowohl möglich, daß es keine, als auch, daß es mehrere Lösungen gibt. Wenn die Differenzen aufeinanderfolgender Ergebnisse von RATE nach 20 Iterationsschritten nicht gegen 0,0000001 konvergieren, liefert RATE den Fehlerwert #NUM!.

`RATE(zzr; rmz; bw; zw; f; schätzwert)`

zzr gibt an, über wie viele Perioden die jeweilige Annuität (Rente) gezahlt wird. (zzr = Anzahl der Zahlungszeiträume)

rmz ist der Betrag (Annuität), der in jeder Periode gezahlt wird. Dieser Betrag bleibt während der Laufzeit konstant. Üblicherweise umfaßt rmz das Kapital und die Zinsen, nicht jedoch sonstige Gebühren oder Steuern. (rmz = Regelmäßige

Zahlung)

bw ist der Barwert: der Gesamtbetrag, den eine Reihe zukünftiger Zahlungen zum gegenwärtigen Zeitpunkt wert ist.

zw ist der zukünftige Wert (Endwert) oder der Kassenbestand, den Sie nach der letzten Zahlung erreicht haben möchten. Fehlt das Argument *zw*, wird es als 0 angenommen (beispielsweise ist der Endwert eines Darlehens gleich 0).

f kann den Wert 0 oder 1 annehmen und gibt an, wann die Zahlungen fällig sind.

0 Am Ende einer Periode

1 Am Anfang einer Periode

schätzwert entspricht einer Schätzung, wie hoch der Zinssatz sein wird.

Wird kein Schätzwert angegeben, wird er mit 10% veranschlagt.

Wenn RATE nicht konvergiert, sollten Sie einen anderen Schätzwert angeben. Normalerweise konvergiert RATE, wenn der Schätzwert zwischen 0 und 1 liegt.

REPLACE

Ersetzt einen Teil eines Strings durch einen anderen String.

REPLACE(*orgtext*; *startpos*; *anzzei*; *erstext*)

orgtext ist der originale Text

startpos ist die Position im originalen Text, ab der ersetzt werden soll

anzzei ist die Anzahl der Zeichen, die ersetzt werden sollen

erstext ist der ersetzende Text

REPT

Ersetzt einen Teil eines Strings durch einen anderen String.

`REPLACE(orgtext;startpos;anzzei;erstext)`

orgtext ist der originale Text

startpos ist die Position im originalen Text, ab der ersetzt werden soll

anzzei ist die Anzahl der Zeichen, die ersetzt werden sollen

erstext ist der ersetzende Text

RIGHT

Liefert das letzte Zeichen eines Strings.

`RIGHT(text[; anz])`

text ist der gewählte String

anz ist die Anzahl der Buchstaben, die zurückgeliefert werden.

ROUND

Rundet eine Zahl auf eine bestimmte Anzahl an Dezimalstellen.

`ROUND(zahl; anz)`

zahl ist die Zahl, die gerundet werden soll.

anz gibt an, auf wie viele Dezimalstellen gerundet werden soll.

Ist *anz* größer als 0, wird Zahl auf die angegebene Anzahl an Dezimalstellen gerundet.

Ist *anz* gleich 0, wird Zahl auf die nächste ganze Zahl gerundet.

Ist *anz* kleiner als 0, wird der links vom Dezimalzeichen stehende Teil von Zahl gerundet.

ROW

liefert die Zeilennummer

ROW(x)

x Verweis auf Zeile

ROWS

liefert die Anzahl der Zeilen im Bereich

ROWS(x)

x Verweis auf Bereich

SEARCH

Liefert die Position des ersten Zeichens eines zu suchenden Strings in einem anderen String.

SEARCH(suchtext; text[; startpos])

suchtext ist der zu suchende Text.

text ist der String in dem der Text steht.

startpos ist die Position im Text ab der mit dem Suchen begonnen wird.

Eine als *suchtext* angegebene Zeichenfolge darf die Platzhalterzeichen

Fragezeichen (?) und Stern (*) enthalten. Ein Fragezeichen ersetzt ein Zeichen; ein Sternchen ersetzt eine beliebige Zeichenfolge. Suchen Sie nach einem Fragezeichen oder Sternchen, müssen Sie eine Tilde (~) vor das zu suchende Zeichen setzen. Wird die von Suchtext angegebene Zeichenfolge nicht gefunden, liefert die Funktion den Fehlerwert #VALUE!.

SECOND	<p>gibt die Sekunde aus einem Datumswert zurück</p> <p>SECOND(x)</p> <p>x ist die Zeit in der Form hh:mm:ss als Text oder ein Zahlenwert.</p>
SIGN	<p>Ermittelt das Vorzeichen einer Zahl.</p> <p>SIGN(zahl)</p> <p>liefert 1, wenn die Zahl positiv ist. liefert 0, wenn die Zahl Null ist. liefert -1, wenn die Zahl negativ ist.</p>
SIN	<p>liefert den Sinus einer Zahl</p> <p>SIN(x)</p>
SINH	<p>liefert den Sinus Hyperbolicus einer Zahl</p> <p>SINH(x)</p>
SLN	<p>Liefert die lineare Abschreibung eines Wirtschaftsgutes pro Periode.</p> <p>SLN(wert; restwert; nutzungsdauer)</p> <p><i>wert</i> sind die Anschaffungskosten eines Wirtschaftsgutes.</p> <p><i>restwert</i> ist der Restwert am Ende der Nutzungsdauer.</p> <p><i>nutzungsdauer</i> ist die Anzahl der Perioden, über die das Wirtschaftsgut abgeschrieben wird.</p>

SQRT	<p>Liefert die Quadratwurzel einer Zahl</p> <p><code>SQRT(zahl)</code></p>
STDEV	<p>Schätzt die Standardabweichung ausgehend von einer Stichprobe.</p> <p><code>STDEV(zahl1; zahl2;...)</code></p> <p><i>zahl1, zahl2,...</i> sind 1 bis 30 numerische Argumente, die eine, aus einer Grundgesamtheit gezogene, Stichprobe darstellen.</p> <p>Anstelle der durch Semikola voneinander getrennten Argumente können Sie eine Matrix oder einen Bezug auf eine Matrix angeben.</p>
STDEVP	<p>Berechnet die Standardabweichung ausgehend von der Grundgesamtheit. Es wird vorausgesetzt, daß alle Werte als Argumente gegeben wird.</p> <p><code>STDEVP(zahl1; zahl2;...)</code></p> <p><i>zahl1, zahl2,...</i> sind 1 bis 30 numerische Argumente, die einer Grundgesamtheit entsprechen. Anstelle der durch Semikola voneinander getrennten Argumente können Sie eine Matrix oder einen Bezug auf eine Matrix angeben.</p>
SUBSTITUTE	<p>Verändert einen String, in dem ein Teilstring durch einen anderen String ersetzt wird.</p> <p><code>SUBSTITUTE(text; altertext; neuertext)</code></p> <p><i>text</i> ist ein String der <i>altertext</i> enthält.</p> <p><i>altertext</i> ist ein Teilstring von <i>text</i>.</p>

neuertext ist der einzusetzende String.

SUM	Liefert die Summe der Argumente. <code>SUM(zahl1; zahl2;...)</code>
SUMSQ	Liefert die Summe aller Quadrate der Argumente. <code>SUMSQ(zahl1; zahl2;...)</code>
SYD	Liefert die arithmetisch-degressive Abschreibung eines Wirtschaftsgutes für eine bestimmte Periode. Anmerkung! Diese Funktion ist nur bedingt auf das deutsche Steuerrecht anwendbar. <code>SYD(wert; restwert; nutzungsdauer; zr)</code> <i>wert</i> sind die Anschaffungskosten eines Wirtschaftsgutes. <i>restwert</i> ist der Restwert am Ende der Nutzungsdauer. <i>nutzungsdauer</i> ist die Anzahl der Perioden, über die das Wirtschaftsgut abgeschrieben wird. <i>zr</i> ist die Periode und muß dieselbe Zeiteinheit verwenden wie die Nutzungsdauer.
T	Testet den Wert. Ist dieser ein String, so wird er zurückgeliefert. Ansonsten wird ein leerer String zurückgeliefert. <code>T(wert)</code>
TAN	liefert den Tangens einer Zahl

TAN(x)

TANH liefert den Tangens Hyperbolicus einer Zahl

TANH(x)

TEXT Diese Funktion wandelt einen Wert unter Berücksichtigung eines speziellen Zahlenformats in einen Text um.

TEXT(wert; format)

wert ist ein numerischer Wert, eine Formel, die einen numerischen Wert liefert, oder ein Bezug auf eine Zelle, die einen numerischen Wert enthält.

format ist ein als Zeichenfolge angegebenes Zahlenformat. Das Format muß in Anführungsstrichen gesetzt werden. Es können verschiedene Formatstrings verwendet werden, inklusive "General", "dd.mm.yyyy" und "hh:mm:ss".

format darf weder Sternchen (*) enthalten noch mit "Standard" belegt sein.

TIME Liefert die fortlaufende Zahl einer bestimmten Uhrzeit. Eine von der Funktion TIME gelieferte fortlaufende Zahl ist ein Dezimalbruch im Bereich von 0 bis 0,99999999 und entspricht einer Uhrzeit von 0:00:00 (24:00:00) bis 23:59:59.

TIME(stunde; minute; sekunde)

stunde ist eine Zahl von 0 bis 23, die für die Stunde steht.

minute ist eine Zahl von 0 bis 59, die für die Minute steht.

sekunde ist eine Zahl von 0 bis 59, die für die Sekunde steht.

TIMEVALUE Liefert die fortlaufende Zahl einer bestimmten Uhrzeit, die im Textformat vorliegt. Eine von der Funktion TIMEVALUE gelieferte fortlaufende Zahl ist ein Dezimalbruch im Bereich von 0 bis 0,99999999 und entspricht einer Uhrzeit von 0:00:00 (24:00:00) bis 23:59:59.

TIMEVALUE(text)

text ist eine Zeitangabe z.B. "23:45".

TODAY Liefert das aktuelle Datum als Zahl.

TODAY()

TRIM entfernt alle Leerzeichen in einem String mit Ausnahme eines Leerzeichens zwischen Wörtern.

TRIM(text)

text kann jeder String oder eine Referenz auf eine Zelle sein, die einen Text enthält.

TRUE gibt den logischen Wert TRUE(Wahr) zurück

TRUE()

TRUNC Diese Funktion verkürzt eine Zahl zu einer ganzen Zahl, indem sie die Nachkommastellen der Zahl abschneidet.

TRUNC(zahl[; anz])

zahl ist die Zahl, deren Stellen Sie abschneiden

möchten.

anz ist eine Zahl, die angibt, wie viele Nachkommastellen erhalten bleiben sollen. Die Voreinstellung für *anz* ist 0.

TYPE

Liefert den Typ von Wert. Die Funktion TYPE können Sie immer dann einsetzen, wenn die weitere Vorgehensweise einer Funktion davon abhängt, welchen Typ der in einer bestimmten Zelle enthaltene Wert hat.

TYPE(wert)

wert kann ein beliebiger Wert sein, beispielsweise eine Zahl, ein Text, ein Wahrheitswert usw..

Zahl	1
Text	2
Wahrheitswert	4
Formel	8
Fehlerwert	16
Matrix	64

UPPER

Wandelt die Buchstaben in dem ausgewählten String in Großbuchstaben.

UPPER(text)

VALUE

Diese Funktion wandelt Text in eine Zahl um.

VALUE(text)

text gibt die Zeichenfolge an, die Sie umwandeln möchten, und steht entweder in Anführungszeichen oder ist ein Bezug auf die Zelle, die den Text enthält. Text kann in jedem der Formate vorliegen, die Mein System bezüglich konstanter Zahlen, Tagesdaten und

Uhrzeiten unterstützt. Liegt Text in keinem dieser Formate vor, liefert die Funktion VALUE den Fehlerwert #VALUE!.

VAR

Schätzt die Varianz, ausgehend von einer Stichprobe.

`VAR(zahl1; zahl2; ...)`

zahl1, zahl2,... sind 1 bis 30 numerische Argumente, die eine, aus einer Grundgesamtheit gezogene, Stichprobe darstellen.

VARP

Berechnet die Varianz, ausgehend von der Grundgesamtheit.

`VARP(zahl1; zahl2; ...)`

zahl1, zahl2,... sind 1 bis 30 numerische Argumente, die einer Grundgesamtheit entsprechen.

VDB

Diese Funktion berechnet für den von Ihnen angegebenen Abschreibungszeitraum, einschließlich Teilperioden, den für ein Anlagegut zu berücksichtigenden Abschreibungsbetrag. Die Berechnung erfolgt nach der Methode der degressiven Doppelraten-Abschreibung oder einer anderen, von Ihnen angegebenen Methode. VDB steht für Variable Declining Balance (Variable degressive Abschreibung oder variable geometrisch-degressive Abschreibung).

`VDB(wert; restwert; nutzungsdauer; anfang; ende; faktor; nwechseln)`

wert sind die Anschaffungskosten eines Wirtschaftsgutes.

restwert ist der Restwert am Ende der Nutzungsdauer.

nutzungsdauer ist die Anzahl der Zeiträume, über die

das Wirtschaftsgut abgeschrieben wird.

anfang ist der erste Zeitraum, der für die Berechnung eines Abschreibungsbetrags berücksichtigt werden soll. *anfang* muß in derselben Zeiteinheit vorliegen wie *nutzungsdauer*.

ende ist der letzte Zeitraum, der für die Berechnung eines Abschreibungsbetrages berücksichtigt werden soll. *ende* muß in derselben Zeiteinheit vorliegen wie *nutzungsdauer*.

faktor ist das Maß, um das die Abschreibung abnimmt. Fehlt das Argument *faktor*, wird es als 2 angenommen (das Verfahren der degressiven Doppelraten-Abschreibung). Wenn Sie das Verfahren der degressiven Doppelraten-Abschreibung nicht anwenden möchten, müssen Sie einen anderen Faktor angeben. Eine Beschreibung des Verfahrens der degressiven Doppelraten-Abschreibung finden Sie unter der Funktion DDB.

nwechseln ist ein Wahrheitswert, der angibt, ob auf lineare Abschreibung umgeschaltet werden soll, wenn der dabei berechnete Abschreibungsbetrag größer ist als der bei der geometrischen Abschreibung.

Ist *nwechseln* mit TRUE belegt, wechselt Mein System selbst dann nicht zu dem Verfahren der linearen Abschreibung, wenn der dabei berechnete Abschreibungsbetrag größer ist als der bei der geometrischen Abschreibung. Ist *nwechseln* mit FALSE belegt oder nicht angegeben, wechselt Mein System zu dem Verfahren der linearen Abschreibung, wenn der dabei berechnete Abschreibungsbetrag größer ist als der bei der geometrischen Abschreibung.

Mit Ausnahme von *nwechseln* müssen alle Argumente positive Zahlen sein.

VLOOKUP

Prüft, ob die erste Spalte einer Matrix einen bestimmten Wert enthält, und liefert dann den Wert, der in der angegebenen Zelle steht. VLOOKUP sollten Sie immer dann anstelle von HLOOKUP verwenden, wenn Ihre Vergleichswerte in einer Spalte stehen, die sich links oder rechts von den Daten befindet, nach denen Sie suchen.

VLOOKUP(suchkriterium; matrix; spaltenindex; bereich_verweis)

suchkriterium ist der Wert, nach dem Sie in der ersten Spalte der Matrix suchen. *suchkriterium* kann ein Wert, ein Bezug oder eine Zeichenfolge (Text) sein.

matrix ist die Informationstabelle, in der Daten gesucht werden. Verwenden Sie einen Bezug auf einen Bereich oder einen Bereichsnamen, wie z.B. "Datenbank" oder "Liste".

Ist *bereich_verweis* gleich TRUE, müssen die in der ersten Spalte von Matrix stehenden Werte in aufsteigender Reihenfolge angeordnet sein: ..., -2, -1, 0, 1, 2, ... , A-Z, FALSE, TRUE, andernfalls liefert VLOOKUP eventuell ein falsches Ergebnis. Ist *bereich_verweis* gleich FALSE ist es nicht notwendig, daß die Werte sortiert vorliegen.

Sie können die Werte in aufsteigende Reihenfolge bringen, indem Sie den Schalter Sortieren verwenden.

Die in der ersten Spalte von *matrix* stehenden Werte dürfen Zeichenfolgen (Texte), Zahlen oder Wahrheitswerte sein.

Zeichenfolgen (Texte), die sich nur bezüglich der Schreibweise (Klein-/Großbuchstaben) unterscheiden, werden als gleich angesehen.

spaltenindex ist die Nummer der Spalte in der

Mehrfachoperationsmatrix, aus der der übereinstimmende Wert geliefert werden soll. Ein Spaltenindex von 1 liefert den Wert, der in der ersten Spalte von *matrix* steht; ein Spaltenindex von 2 liefert den Wert, der in der zweiten Spalte von *matrix* steht usw. Ist *spaltenindex* kleiner als 1, liefert VLOOKUP den Fehlerwert #VALUE!, ist *spaltenindex* größer als die Anzahl der zu *matrix* gehörenden Spalten, liefert VLOOKUP den Fehlerwert #REF!.

bereich_verweis ist ein Wahrheitswert, der angibt, wie die zu Matrix gehörenden Werte sortiert sind. Ist *bereich_verweis* gleich TRUE oder nicht angegeben, müssen die Werte, die die erste Spalte von *matrix* bilden, in aufsteigender Reihenfolge vorliegen. Ist *bereich_verweis* gleich FALSE, spielt es keine Rolle, in welcher Reihenfolge die zu Matrix gehörenden Werte angeordnet sind.

WEEKDAY

Diese Funktion liefert den Tag der Woche, der dem Argument *Zahl* entspricht. Der Tag wird als ganze Zahl ausgegeben, die einen Wert von 1 (Sonntag) bis 7 (Samstag) annehmen kann.

WEEKDAY(*zahl*)

zahl ist eine Kodierung für Datum und Zeit, die Mein System für Datums- und Zeitberechnungen verwendet. Sie müssen das Argument *zahl* nicht als Zahl, sondern können es auch als Text angeben, also zum Beispiel "15. April 1993" oder "15.4.93". Ein solcher Text wird automatisch in die entsprechende fortlaufende Zahl umgewandelt.

YEAR

Diese Funktion liefert das Jahr, das dem Argument *Zahl* entspricht. Das Jahr wird als ganze Zahl

ausgegeben.

YEAR(zahl)

zahl ist die Kodierung für Datum und Zeit, die Mein System für Datums- und Zeitberechnungen verwendet. Sie müssen das Argument *zahl* nicht als Zahl, sondern können es auch als Text angeben, also zum Beispiel "15. April 1993" oder "15.4.93". Ein solcher Text wird automatisch in die entsprechende fortlaufende Zahl umgewandelt.