

@Funktionen A-Z

A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z

A

@(Adresse)

Ermittelt indirekt den Inhalt der Zelle, die in *Adresse* angegeben wurde.

@ABKÜRZEN(x;[n])

Schneidet *x* auf eine angegebene Anzahl von Dezimalstellen *n* ab.

@ABRUNDEN(x;[n];[Richtung])

Rundet einen Wert *x* auf das nächste Vielfache der angegebenen Zehnerpotenz *n* ab.

@ABS(x)

Berechnet den absoluten (positiven) Wert von *x*.

@ABWQUADRAT(Argumentliste)

Berechnet die Summe der quadratischen Abweichung in *Argumentliste*.

@AFADEG(Kosten;Restwert;Lebensdauer;Zeitraum)

Berechnet die Abschreibung eines Anlagegutes mit der doppelten degressiven Abschreibung.

@AFADEGV(Kosten;Restwert;Lebensdauer;Startperiode;Endperiode;[Abschreibung];[Schalter])

Berechnet die Abschreibung eines Anlagegutes mit der variablen degressiven Abschreibung.

@AFADIG(Kosten;Restwert;Lebensdauer;Zeitraum)

Berechnet die digitale Abschreibung eines Anlagegutes für eine angegebene *Periode*.

@AFAEDEG(Kosten;Restwert;Lebensdauer;Zeitraum)

Berechnet die Abschreibung eines Anlagegutes mit der festen degressiven Abschreibung.

@AFALIN(Kosten;Restwert;Lebensdauer)

Berechnet die lineare Abschreibung eines Anlagegutes mit anfänglichen *Kosten*, der erwarteten *Lebensdauer* und einem *Restwert* für eine Periode.

@AKTWERT2(Zahlungen;Zinssatz;Laufzeit)

Berechnet den aktuellen Wert einer Investition, basierend auf einer Reihe gleicher *Zahlungen* unter Diskontierung eines periodischen *Zinssatzes* über die Anzahl der Zahlungsperioden in *Laufzeit* bei nachschüssiger Rente.

@AKTWERT(Zahlungen;Zinssatz;Laufzeit)

Berechnet den aktuellen Wert einer Investition, basierend auf einer Reihe gleicher *Zahlungen* unter Diskontierung eines periodischen *Zinssatzes* über die Anzahl der Zahlungsperioden in *Laufzeit* bei nachschüssiger Rente.

@ANN2(Zahlungen;Zinssatz;Zukwert)

Berechnet die Zahl der erforderlichen Perioden gleicher *Zahlungen*, um bei gleichbleibendem *Zinssatz* und vorschüssiger Rente einen bestimmten *Zukwert* anzusparen.

@ANN(Zahlungen;Zinssatz;Zukwert)

Berechnet die Zahl der erforderlichen Perioden für eine Reihe gleicher *Zahlungen*, um bei gleichbleibendem *Zinssatz* und nachschüssiger Rente einen bestimmten *Zukwert* anzusparen.

@ANNUITÄT(Kapital;Zinssatz;Laufzeit;[Typ];[Zukwert])

Berechnet die Rückzahlung eines Darlehens (*Kapital*) bei einem gegebenen *Zinssatz* für die angegebene Anzahl an Zahlungsperioden (*Laufzeit*).

@ANZAHL(Argumentliste)

Zählt die nicht-leeren Zellen in *Argumentliste* mit Bereichen.

@ARBEITSTAG(Startdatum;Tage;[Freie-Tage];[Wochenenden])

Berechnet die Datumseriennummer für ein Datum, das eine angegebene Zahl von *Tagen* vor oder nach dem *Startdatum* liegt.

@ARCCOSEC(x)

Berechnet den Arkuskosekans unter Verwendung des Kosekans x eines Winkels.

@ARCCOS(x)

Berechnet den Arkuskosinus unter Verwendung des Kosinus x eines Winkels.

@ARCCOT(x)

Berechnet den Arkuskotangens unter Verwendung des Kotangens x eines Winkels.

@ARCOSECH(x)

Berechnet den Areakosekans unter Verwendung des hyperbolischen Kosekans x eines Winkels.

@ARCOSH(x)

Berechnet den Areakosinus unter Verwendung des hyperbolischen Kosinus x eines Winkels.

@ARCOTH(x)

Berechnet den Areakotangens unter Verwendung des hyperbolischen Kotangens x eines Winkels.

@ARCSEC(x)

Berechnet den Arkussekans unter Verwendung des Sekans x eines Winkels.

@ARCSIN(x)

Berechnet den Arkussinus unter Verwendung des Sinus x eines Winkels.

@ARCTAN2(x;y)

Berechnet den Arkustangens unter Verwendung des Tangens y/x eines Winkels. Das Ergebnis von @ARCTAN2 ist ein Winkel in Radiant von $-\pi$ bis π , abhängig von den Vorzeichen von x und y .

@ARCTAN(x)

Berechnet den Arkustangens unter Verwendung des Tangens x eines Winkels. Das Ergebnis von @ARCTAN ist ein Winkel in Radiant, von $-\pi/2$ bis $\pi/2$.

@ARTANH(x)

Berechnet den Areatangens unter Verwendung des hyperbolischen Tangens x eines Winkels.

@ARSECH(x)

Berechnet den Areasekans unter Verwendung des hyperbolischen Sekans x eines Winkels.

@ARSINH(x)

Berechnet den Areasinus unter Verwendung des hyperbolischen Sinus x eines Winkels.

@AUFRUNDEN(x;[n];[Richtung])

Rundet einen Wert x auf das nächste Vielfache der angegebenen Zehnerpotenz n auf.

@AWERT(Zahlungen;Zinssatz;Laufzeit;[Typ];[Zukwert])

Berechnet den aktuellen Wert einer Kapitalanlage mit einem angegebenen *Zukwert* bei vorschüssiger oder nachschüssiger Rente.

@AWERTBETRAG(Zukwert;Zinssatz;Laufzeit;[Zinstermine])

Berechnet den aktuellen Wert eines angelegten Pauschalbetrags für eine gegebene Anzahl von zukünftigen Perioden (*Laufzeit*) bei Diskontierung eines gegebenen *Zinssatzes*.

B

@BEREICHNAME(Zelle)

Gibt den Namen des Bereichs zurück, in dem sich *Zelle* befindet.

@BESSELI(x;n)

Berechnet die modifizierte Bessel-Funktion ganzzahliger Ordnung $I_n(x)$.

@BESSELJ(x;n)

Berechnet die Bessel-Funktion ganzzahliger Ordnung $J_n(x)$.

@BESSELK(x;n)

Berechnet die geänderte Bessel-Funktion ganzzahliger Ordnung $K_n(x)$.

@BESSELY(x;n)

Berechnet die Bessel-Funktion ganzzahliger Ordnung $Y_n(x)$, auch bekannt als Neumann-Funktion.

@BETA(z;w)

Berechnet die Beta-Funktion.

@BETA(a;b;x)

Berechnet die unvollständige Beta-Funktion.

@BINOMIAL(Versuche;Ergebnisse;Wahrscheinlichkeit;[Typ])

Berechnet die binomiale Wahrscheinlichkeits-Massenfunktion oder die kumulative Binomialverteilung.

@BLÄTTER(Bereich)

Zählt die Anzahl der Blätter in *Bereich*.

C

@CHITEST(Bereich1;[Bereich2])

Führt einen Chi-Quadrat-Test auf Unabhängigkeit mit den Daten in *Bereich1* oder einen Chi-Quadrat-Test auf Güte der Anpassung mit den Daten in *Bereich1* und *Bereich2* durch.

@CHIVERTeilUNG(x;Freiheitsgrad;[Typ])

Berechnet die Chi-Quadrat-Verteilung.

@CODE(Zeichenfolge)

Liefert den Code des LMBCS-Zeichensatzes (Lotus Multibyte Character Set) zurück, der dem ersten Zeichen in *Zeichenfolge* entspricht.

@COS(x)

Berechnet den Kosinus eines Winkels x . Der Winkel muß in Radiant gemessen sein.

@COSEC(x)

Berechnet den Kosekans eines Winkels x . Der Winkel muß in Radiant gemessen sein.

@COSECH(x)

Berechnet den hyperbolischen Kosekans eines Winkels x . Der Winkel muß in Radiant gemessen sein.

@COSH(x)

Berechnet den hyperbolischen Kosinus eines Winkels x . Das Ergebnis von @COSH ist größer oder gleich 1. Der Winkel muß in Radiant gemessen sein.

@COT(x)

Berechnet den Kotangens eines Winkels x . Der Winkel muß in Radiant gemessen sein.

@COTH(x)

Berechnet den hyperbolischen Kotangens eines Winkels x . Der Winkel muß in Radiant gemessen sein.

D

@D360(Anfangsdatum;Enddatum)

Berechnet die Anzahl von Tagen zwischen zwei Datumsangaben basierend auf einem Jahr mit 360 Tagen.

@DANZAHL(Datenbank-Bereich;Feld;[Kriterien-Bereich])

Zählt die nicht-leeren Zellen in einem *Feld* einer Datenbanktabelle, die dem angegebenen *Kriterien-Bereich* entsprechen.

@DATALINK(Anwendung;Thema;Element;[Format];[Max-Zeilen];[Max-Spalten];[Max-Blätter])

Erstellt eine OLE-Verknüpfung zu Daten.

@DATUM(Jahr;Monat;Tag)

Berechnet die Datumseriennummer für das angegebene *Jahr*, den *Monat* und den *Tag*.

@DATUMDIFF(Anfangsdatum;Enddatum;Format)

Berechnet die Anzahl von Jahren, Monaten oder Tagen zwischen *Anfangsdatum* und *Enddatum*.

@DATUMFOLGE(Datum)

Konvertiert eine Datumseriennummer in das entsprechende Datum und zeigt es unter Verwendung des internationalen Standard-Datumsformats als Label an.

@DATUMINFO(Datum;Attribut)

Liefert Informationen über ein Datum.

@DATUMKONV(Datum;Eingabetyp;Ausgabetyp)

Konvertiert ein Hijri (arabisches), Farsi (iranisches) oder ein hebräisches (israelisches) Datum in ein gregorianisches Datum bzw. umgekehrt. Ist nur für Versionen von 1-2-3 verfügbar, die bidirektionale Sprachen unterstützen.

@DATUMWERT(Zeichenfolge)

Berechnet die Datumseriennummer für das Datum in Zeichenfolge.

@DAYS360(Anfangsdatum;Enddatum)

Berechnet die Anzahl von Tagen zwischen Anfangsdatum und Enddatum, basierend auf einem Jahr mit 360 Tagen entsprechend den Standards des amerikanischen Wertpapierhandels.

@DBMAX(Datenbank-Bereich;Feld;[Kriterien-Bereich])

Findet den höchsten Wert in einem Feld in einer Datenbanktabelle, der dem angegebenen Kriterien-Bereich entspricht.

@DBMIN(Datenbank-Bereich;Feld;[Kriterien-Bereich])

Findet den kleinsten Wert in einem Feld in einer Datenbanktabelle, der dem angegebenen Kriterien-Bereich entspricht.

@DBVARIANZ(Datenbank-Bereich;Feld;[Kriterien-Bereich])

Berechnet die Varianz der Grundgesamtheit der Werte in einem Feld in einer Datenbanktabelle, die dem angegebenen Kriterien-Bereich entsprechen.

@DEZIL(Teil;Bereich)

Gibt ein angegebenes Dezil zurück.

@DEZIMAL(Hex_Zeichenfolge)

Konvertiert einen hexadezimalen Wert in sein dezimales Äquivalent.

@DHOLEN(Datenbank-Bereich;Feld;[Kriterien-Bereich])

Holt einen Wert bzw. ein Label aus einem Feld in einer Datenbanktabelle, der bzw. das dem angegebenen Kriterien-Bereich entspricht.

@DMITTELWERT(Datenbank-Bereich;Feld;[Kriterien-Bereich])

Berechnet den Durchschnitt der Werte in einem Feld in einer Datenbanktabelle, die dem angegebenen Kriterien-Bereich entsprechen.

@DPURANZAHL(Datenbank-Bereich;Feld;[Kriterien-Bereich])

Zählt die Zellen in einem Feld einer Datenbanktabelle, die Werte enthalten, die dem angegebenen Kriterien-Bereich entsprechen.

@DSTDABW(Datenbank-Bereich;Feld;[Kriterien-Bereich])

Berechnet die Standardabweichung einer Grundgesamtheit von Werten in einem Feld in einer Datenbanktabelle, die dem angegebenen Kriterien-Bereich entsprechen.

@DSTDABWP(Datenbank-Bereich;Feld;[Kriterien-Bereich])

Berechnet die Standardabweichung von Stichproben von Werten in einem Feld in einer Datenbanktabelle, die dem angegebenen Kriterien-Bereich entsprechen.

@DSUMME(Datenbank-Bereich;Feld;[Kriterien-Bereich])

Berechnet die Summe der Werte in einem Feld in einer Datenbanktabelle, die dem angegebenen Kriterien-Bereich entsprechen.

@DVARP(Datenbank-Bereich;Feld;[Kriterien-Bereich])

Berechnet die Varianz von Stichproben der Werte in einem Feld in einer Datenbanktabelle, die dem angegebenen Kriterien-Bereich entsprechen.

E

@EFFBELA(Laufzeit;Zahlung;Aktwert;[Typ];[Zukwert];[Schätzwert])

Berechnet den periodischen Zinssatz, bei dem eine Investition mit dem (Aktwert) über die angegebene Laufzeit auf einen Zukwert anwächst.

@EIGENNAME(Zeichenfolge)

Schreibt den ersten Buchstaben in jedem Wort in Zeichenfolge groß und wandelt die restlichen Buchstaben in

Kleinbuchstaben um.

@ENDSUMME(*Argumentliste*)

Berechnet die Summe aller Zellen in *Argumentliste*, die die Funktion @ZWISCHENSUMME in einer Formel enthalten.

@ERSETZEN(*Original_Zeichenfolge*; *Startnummer*; *n*; *Neue-Zeichenfolge*)

Ersetzt *n* Zeichen in *Original-Zeichenfolge* mit der *Neuen-Zeichenfolge*, beginnend bei *Startnummer*.

@ERSETZENB(*Original_Zeichenfolge*; *Startnummer*; *n*; *Neue-Zeichenfolge*)

Ersetzt *n* Byte in *Original-Zeichenfolge* mit der *Neuen-Zeichenfolge*, beginnend bei *Startnummer*.

@EXP(*x*)

Berechnet den Wert der Konstanten *e* (ca. 2,718282) hoch *x* erhoben.

@EXP2(*x*)

Berechnet den Wert der Konstanten *e* (ca. 2,718282) hoch $(-x^2)$ erhoben.

@EZIFFERNFOLGE(*Ziffernfolge*)

Konvertiert eine *Ziffernfolge* von Thai-numerischen Zeichen in eine arabische Zahl.

F

@FAKULTÄT(*n*)

Berechnet die Fakultät von *n*.

@FAKULTÄTLN(*n*)

Berechnet den natürlichen Logarithmus der Fakultät von *n*.

@FALSCH

Gibt den logischen Wert 0 (falsch) zurück.

@FEHLER

Gibt den Wert FEHLER zurück.

@FEHLERF(*Untergrenze*; [*Obergrenze*])

Berechnet die Fehlerfunktion bei angegebener *Obergrenze* und *Untergrenze*.

@FEHLERFD(*x*)

Berechnet die Ableitung der Fehlerfunktion.

@FEHLERFK(*x*)

Berechnet die komplementäre Fehlerfunktion zwischen *x* und ∞ (unendlich).

@FINDEN(*Such-Zeichenfolge*; *Zeichenfolge*; *Startnummer*)

Berechnet die Position in *Zeichenfolge*, an der 1-2-3 das erste Vorkommen von *Such-Zeichenfolge* findet, beginnend bei *Startnummer*.

@FINDENB(*Such-Zeichenfolge*; *Zeichenfolge*; *Startnummer*)

Berechnet die Byte-Position in *Zeichenfolge*, an der 1-2-3 das erste Vorkommen von *Such-Zeichenfolge* findet, beginnend bei *Startnummer*.

@FOLGE(*x*; *n*)

Wandelt einen Wert *x* in ein Label um und verwendet dazu das durch *n* angegebene Format.

@FTEST(*Bereich1*; *Bereich2*)

Führt einen *F*-Test durch und ermittelt die entsprechende Wahrscheinlichkeit.

@FULLP(*Label*)

Wandelt Ein-Byte-Zeichen (ASCII) in *Label* in die entsprechenden japanischen Zwei-Byte-Zeichen um.

@FVERTEILUNG(*x*; *Freiheitsgrad1*; *Freiheitsgrad2*; [*Typ*])

Berechnet the *F*-Verteilung.

G

@GAMMA(*x*)

Berechnet die Gamma-Funktion.

@GAMMAI(*a*; *x*; [*Komplement*])

Berechnet die unvollständige Gamma-Funktion.

@GAMMALN(*x*)

Berechnet den natürlichen Logarithmus der Gamma-Funktion.

@GEOMITTEL(*Argumentliste*)

Berechnet das geometrische Mittel der Werte in *Argumentliste*.

@GERADE(x)

Rundet einen Wert *x* zur nächsten geraden Ganzzahl auf.

@GEWMITTELWERT(Datenbereich;Gewichtungs_Bereich;[Typ])

Berechnet den gewichteten Mittelwert der Werte in *Datenbereich*.

@GLEICH(Zeichenfolge1;Zeichenfolge2)

Gibt eine 1 (wahr) zurück, wenn *Zeichenfolge1* und *Zeichenfolge2* exakt gleich sind; gibt andernfalls eine 0 (falsch) zurück.

@GRADINRAD(Grad)

Konvertiert *Grad* in Radiant.

@GROSS(Zeichenfolge)

Setzt alle Buchstaben in einer *Zeichenfolge* in Großbuchstaben um.

H

@HALFP(Label)

Wandelt japanische Zwei-Byte-Zeichen in *Label* in die entsprechenden Ein-Byte-Zeichen (ASCII) um.

@HARMITTEL(Argumentliste)

Berechnet das harmonische Mittel der Werte in *Argumentliste*.

@HEUTE

Berechnet die Datumseriennummer für das aktuelle Systemdatum.

@HEX(x)

Wandelt eine Dezimalzahl in die entsprechende Hexadezimalzahl um.

@HVERWEIS(x;Bereich;Versatz)

Ermittelt den Inhalt einer angegebenen Zelle in einer horizontalen Verweistabelle, in der durch *x* bezeichneten Spalte.

I

@INDEX(Bereich;Spalte;Zeile;[Blatt])

Ermittelt den Inhalt einer Zelle in einer angegebenen *Spalte*, *Zeile* und einem *Blatt* in einem *Bereich*.

@INFO(Attribute)

Liefert Informationen über die aktuelle 1-2-3 Sitzung.

@INT(x)

Ermittelt den ganzzahligen Teil eines Wertes *x*.

@INTZINS(Schätzwert;Bereich)

Berechnet die interne Ertragsrate für eine Reihe von investitionsbedingten Cash-flows.

@ISTBEREICH(Bereich)

Testet, ob *Bereich* ein definierter Bereichsname oder eine gültige Bereichsadresse ist, und gibt 1 (wahr) oder 0 (falsch) zurück.

@ISTDATEI(Dateiname;[Typ])

Testet, ob es sich bei *Dateiname* um eine gerade geladene Datei oder eine Datei auf der Festplatte handelt, und gibt 1 (wahr) oder 0 (falsch) zurück.

@ISTDEFZUS(Name)

Testet, ob *Name* eine definierte globale LotusScript Funktion ist, und gibt 1 (wahr) oder 0 (falsch) zurück.

@ISTFEHLER(x)

Testet, ob *x* der Wert FEHLER ist, und gibt 1 (wahr) oder 0 (falsch) zurück.

@ISTFOLGE(x)

Testet, ob *x* eine Zeichenfolge oder eine Zelle ist, die ein Label enthält, bzw. eine Formel, die ein Label ergibt, und gibt 1 (wahr) oder 0 (falsch) zurück.

@ISTGELZUS(Name)

Testet, ob es sich bei *Name* um eine gerade geladene Zusatzanwendung handelt, und gibt 1 (wahr) oder 0 (falsch) zurück.

@ISTLEER(Adresse)

Testet, ob *Adresse* eine leere Zelle ist, und gibt 1 (wahr) oder 0 (falsch) zurück.

@ISTMAKRO(Name)

Testet, ob *Name* eine definierte globale LotusScript Subroutine ist, und gibt 1 (wahr) oder 0 (falsch) zurück.

@ISTNV(x)

Testet, ob *x* der Wert NV ist, und gibt 1 (wahr) oder 0 (falsch) zurück.

@ISTZAHL(x)

Testet, ob *x* der Wert NV, FEHLER oder eine leere Zelle ist, und gibt 1 (wahr) oder 0 (falsch) zurück.

@ISTZWISCHEN(Wert;Grenze1;Grenze2;[Typ])

Testet, ob *Wert* zwischen *Grenze1* und *Grenze2* liegt, und gibt 1 (wahr) oder 0 (falsch) zurück.

J

@JAHR(Datumseriennummer;Typ)

Extrahiert den Tag des Monats, eine Ganzzahl zwischen 1 (1. Januar 1900) und 2958465 (31. Dezember 9999), aus *Datumseriennummer*.

@JETZT

Berechnet die Datumseriennummer (ganzzahliger Teil) und die Zeitseriennummer (Dezimalteil), die dem aktuellen Datum und der aktuellen Uhrzeit der Systemuhr entsprechen.

K

@KLÄREN(Zeichenfolge)

Entfernt nichtdruckbare Zeichen aus *Zeichenfolge*.

@KLEIN(Zeichenfolge)

Wandelt alle Buchstaben in *Zeichenfolge* in Kleinbuchstaben um.

@KOMBIN(n;r)

Berechnet den Binomialkoeffizienten für *n* und *r*. Dies entspricht der Anzahl der Möglichkeiten, die *r* aus *n* ausgewählt werden kann, ohne Berücksichtigung der Reihenfolge.

@KOMPR(Zeichenfolge)

Löscht führende, abschließende und aufeinanderfolgende Leerzeichen aus *Zeichenfolge*.

@KOORD(Arbeitsblatt;Spalte;Zeile;Absolut)

Erstellt eine Zellreferenz von Werten, die *Arbeitsblatt*, *Spalte* und *Zeile* entsprechen.

@KORREL(Bereich1;Bereich2)

Berechnet den Korrelationskoeffizienten von Werten in *Bereich1* und *Bereich2*.

@KOVAR(Bereich1;Bereich2;[Typ])

Berechnet die Kovarianz einer Population oder einer Stichprobe der Werte in *Bereich1* und *Bereich2*.

@KRITBINOMIAL(Versuche;Wahrscheinlichkeit;Alpha)

Ermittelt die größte Ganzzahl, für die die kumulative Binomialverteilung kleiner oder gleich *Alpha* ist.

@KURS(Valutierungstermin;Fälligkeit;Coupon;Rendite;[Rückzahlung];[Zinstermine];[Zinstage])

Berechnet den Kurs pro 100 DM Nominalwert für Wertpapiere mit periodischen Zinszahlungen.

@KURS2(Valutierungstermin;Fälligkeit;Coupon;Rendite;[Zinstermin];[Zinstage])

Berechnet den Kurs pro 100 Yen Nominalwert für Wertpapiere mit periodischen Zinszahlungen anhand japanischer Konventionen.

@KURTOSIS(Bereich;[Typ])

Berechnet die Kurtosis der Werte in *Bereich*.

L

@LÄNGE(Zeichenfolge)

Zählt die Zeichen in *Zeichenfolge*.

@LÄNGEB(Zeichenfolge)

Zählt die Anzahl der Bytes in *Zeichenfolge*.

@LAUF(Zinssatz;Zukwert;Aktwert)

Berechnet die Anzahl von Perioden, nach der eine Investition (*Aktwert*) bei festem *Zinssatz* auf einen zukünftigen Wert (*Zukwert*) anwächst.

@LAUFZEIT(Valutierungstermin;Fälligkeit;Coupon;Rendite;[Zinstermine];[Basis])

Berechnet die jährliche Laufzeit für ein Wertpapier mit periodischen Zinszahlungen.

@LINKS(Zeichenfolge;n)

Ermittelt die ersten n Zeichen in *Zeichenfolge*.

@LINKSB(*Zeichenfolge*; n)

Gibt die ersten n Bytes in *Zeichenfolge* zurück.

@LN(x)

Berechnet den natürlichen Logarithmus (Basis e) von x .

@LOG(x)

Berechnet den gewöhnlichen Logarithmus (Basis 10) von x .

M

@MAX(*Argumentliste*)

Ermittelt den größten Wert in *Argumentliste*.

@MAXVERWEIS(*Bereich-Liste*)

Ermittelt die absolute Referenz auf eine Zelle, die den größten Wert in einer Liste mit Bereichen enthält.

@MEDIAN(*Argumentliste*)

Ermittelt den Medianwert in *Argumentliste*.

@MIN(*Argumentliste*)

Ermittelt den kleinsten Wert in *Argumentliste*.

@MINTZINS(*Bereich*; *Finanz_Satz*; *Reinvest_Satz*; [*Typ*])

Berechnet die modifizierte interne Ertragsrate für eine Reihe von investitionsbedingten Cash-flows.

@MINUSZEICHEN(x)

Liefert 1, wenn x ein positiver Wert, 0, wenn x gleich 0, und -1, wenn x ein negativer Wert ist.

@MINUTE(*Zeitseriennummer*)

Extrahiert die Minuten, ein Wert zwischen 0 und 59, aus *Zeitseriennummer*.

@MINVERWEIS(*Bereich-Liste*)

Ermittelt die absolute Referenz auf eine Zelle, die den kleinsten Wert in einer Liste mit Bereichen enthält.

@MITTE(*Zeichenfolge*; *Startnummer*; n)

Kopiert n Zeichen aus *Zeichenfolge* ab dem Zeichen bei *Startnummer*.

@MITTEB(*Zeichenfolge*; *Startnummer*; n)

Kopiert n Bytes aus *Zeichenfolge*, beginnend mit *Startnummer*.

@MITTELABW(*list*)

Berechnet den Mittelwert der absoluten Abweichungen der Werte in *Argumentliste*.

@MITTELWERT(*Argumentliste*)

Berechnet den Mittelwert der absoluten Abweichungen der Werte in *Argumentliste*.

@MLAUFZEIT(*Valutierungstermin*; *Fälligkeit*; *Coupon*; *Rendite*; [*Zinstermine*]; [*Basis*])

Berechnet die modifizierte jährliche Laufzeit für ein Wertpapier mit periodischen Zinszahlungen.

@MOD(x ; y)

Berechnet den Modulus von x/y . Das Vorzeichen des Ergebnisses entspricht dem Vorzeichen von x .

@MODULO(x ; y)

Berechnet den Modulus von x/y . Das Vorzeichen des Ergebnisses entspricht dem Vorzeichen von y .

@MONAT(*Datumseriennummer*)

Extrahiert den Monat, ein Wert zwischen 1 und 12, aus *Datumseriennummer*.

N

@N(*Bereich*)

Gibt den Eintrag in der ersten Zelle eines *Bereichs* als Wert zurück. Wenn die Zelle ein Label enthält, gibt @N den Wert 0 zurück.

@NÄCHSTMONAT(*Startdatum*; *Monate*; [*Tag-des-Monats*]; [*Basis*])

Berechnet die Datumseriennummer für ein Datum, das eine angegebene Zahl von *Monaten* vor oder nach dem *Startdatum* liegt.

@NETAKTWERT(*Zinssatz*; *Bereich*; [*Typ*])

Berechnet den aktuellen Nettowert einer Reihe von zukünftigen Cash-flows in (*Bereich*) unter Diskontierung eines festen periodischen *Zinssatzes*.

@NETTOARBEITSTAGE(*Startdatum*; *Enddatum*; [*Freie-Tage*]; [*Wochenenden*])

Berechnet die Anzahl von Tagen zwischen *Anfangsdatum* und *Enddatum* unter Ausschluß von Ferientagen und Wochenenden.

@NHÖCHST(*Bereich;n*)

Ermittelt den *n*-höchsten Wert in *Bereich*.

@NNIEDRIGST(*Bereich;n*)

Ermittelt den *n*-kleinsten Wert in *Bereich*.

@NPER(*Zahlungen;Zinssatz;Zukwert;[Typ];[Aktwert]*)

Berechnet die Anzahl von Perioden, nach der eine Investition (*Aktwert*) bei festem *Zinssatz* auf einen zukünftigen Wert *Zukwert* anwächst.

@NSUMME(*Versatz;n;Argumentliste*)

Fügt jeden *n*-ten Wert in *Argumentliste* hinzu, beginnend bei *Versatz*.

@NURMEDIAN(*Argumentliste*)

Berechnet den Mittelwert in *Argumentliste* und übergeht dabei alle leeren Zellen, Zellen die Label enthalten oder Zellen mit Formeln, die in Label resultieren.

@NV

Gibt den Wert NV (nicht verfügbar) zurück.

O

P

@PERCENTIL(*x;Bereich*)

Berechnet das *x*-te Perzentil einer Stichprobe der Werte in *Bereich*.

@PERMUT(*n;r*)

Berechnet die Anzahl von Permutationen von *r* Objekten, die aus *n* Objekten ausgewählt werden können.

@PI

Produziert den Wert π (3,14159265358979).

@PMT(*Kapital;Zinssatz;Laufzeit*)

Berechnet die Rückzahlung eines Darlehens (*Kapital*) bei einem gegebenen *Zinssatz* für die angegebene Anzahl an Zahlungsperioden (*Laufzeit*). Diese @Funktion verwendet kanadische Hypothekenkonventionen.

@POISSON(*x;Mittel;[Kumulativ]*)

Berechnet die Poisson-Verteilung.

@PRANG(*x;Bereich;[Stellen]*)

Findet das Perzentil von *x* unter den Werten in *Bereich*.

@PRODUKT(*Argumentliste*)

Multipliziert die Werte in *Argumentliste*.

@PROGNOSE(*x;Y-Bereich;X-Bereich*)

Gibt einen Prognosewert für *x* zurück, basierend auf dem linearen Trend zwischen den Werten im *X-Bereich* und *Y-Bereich*.

@PURANZAHL(*Argumentliste*)

Zählt die Zellen in einer *Argumentliste* von Bereichen, ausgenommen alle Zellen, die Label enthalten.

@PURMAX(*Argumentliste*)

Ermittelt den größten Wert in *Argumentliste*; Zellen mit Labeln werden übergangen.

@PURMIN(*Argumentliste*)

Ermittelt den kleinsten Wert in *Argumentliste*; Zellen mit Labeln werden übergangen.

@PURMITTELWERT(*Argumentliste*)

Berechnet den Mittelwert der Werte in *Argumentliste* und übergeht dabei alle Zellen, die Label enthalten.

@PURSTDABW(*Argumentliste*)

Berechnet die Standardabweichung der Werte in *Argumentliste*, wobei Zellen mit Label übergangen werden.

@PURSTDABWP(*Argumentliste*)

Berechnet die Standardabweichung von Stichproben in *Argumentliste*, wobei Zellen mit Label übergangen werden.

@PURVAR(*Argumentliste*)

Berechnet die Varianz der Grundgesamtheit der Werte in *Argumentliste*, wobei Zellen mit Label übergangen werden.

@PURVARP(*Argumentliste*)

Berechnet die Varianz einer Stichprobe der Werte in *Argumentliste*, wobei Zellen mit Label übergangen werden.

Q

@QUARTIL(*Teil;Bereich*)

Gibt ein angegebenes Quartil zurück.

@QUOTIENT(*x;y*)

Berechnet das Ergebnis von x/y , auf eine Ganzzahl abgeschnitten.

R

@RADINGRAD(*Radiant*)

Wandelt Radiant in Grad um.

@RANG(*Element;Bereich;[Rangfolge]*)

Berechnet die relative Größe oder Position eines Wertes in einem *Bereich* bezogen auf die anderen Werte in dem Bereich.

@RATE(*Kapital;Zinssatz;Laufzeit*)

Berechnet die Rückzahlung eines Darlehens (*Kapital*) bei einem gegebenen *Zinssatz* für die angegebene Anzahl an Zahlungsperioden (*Laufzeit*) bei nachschüssiger Rente.

@RATE2(*Kapital;Zinssatz;Laufzeit*)

Berechnet die Rückzahlung eines Darlehens (*Kapital*) bei einem gegebenen *Zinssatz* für die angegebene Anzahl an Zahlungsperioden (*Laufzeit*) bei vorschüssiger Rente.

@RATEZ(*Kapital;Zinssatz;Laufzeit;Periode*)

Berechnet den *Zinsanteil* einer periodischen Zahlung.

@RECHTS(*Zeichenfolge;n*)

Ermittelt die n letzten Zeichen in *Zeichenfolge*.

@RECHTSB(*Zeichenfolge;n*)

Gibt die letzten n Bytes in *Zeichenfolge* zurück.

@REFKONVERT(*Angabe*)

Konvertiert die Spalten- oder Arbeitsblattbuchstaben von A bis IV in Zahlen von 1 bis 256 und ordnet die Zahlen 1 bis 256 den entsprechenden Buchstaben zu.

@REGRESSION(*X-Bereich;Y-Bereich;Attribute;[Berechnen]*)

Führt eine lineare Mehrfachregression durch und erzeugt die gewünschte Statistik.

@REIHENSUMME(*x;n;m;Koeffizienten*)

Berechnet die Summe einer Potenzreihe.

@RENDITE(*Valutierungstermin;Fälligkeit;Coupon;Kurs;[Rückzahlung];[Zinstermine];[Zinstage]*)

Berechnet die Rendite für Wertpapiere mit periodischen Zinszahlungen.

@RENDITE2(*Valutierungstermin;Fälligkeit;Coupon;Kurs;[Zinstermin];[Zinstage]*)

Berechnet die Rendite für Wertpapiere mit periodischen Zinszahlungen anhand japanischer Konventionen.

@RUNDEN(*x;n*)

Rundet einen Wert x auf das nächste Vielfache der angegebenen Zehnerpotenz n auf.

@RUNDENM(*x;Faktor;[Richtung]*)

Rundet einen Wert x auf den nächsten *Faktor* auf.

S

@S(*Bereich*)

Gibt den Eintrag in der ersten Zelle eines *Bereichs* als Label zurück.

@SCHIEFE(*Bereich;[Typ]*)

Berechnet die Schiefe der Werte in *Bereich*.

@SEC(*x*)

Berechnet den Sekans eines Winkels x .

@SECH(*x*)

Berechnet den hyperbolischen Sekans eines Winkels x .

@SEKUNDE(*Zeitseriennummer*)

Extrahiert die Sekunden, eine Ganzzahl zwischen 0 und 59, aus *Zeitseriennummer*.

@SETZEFOLGE(*Zeichenfolge;Länge;[Ausrichtung]*)

Gibt ein Label zurück, daß *Länge* Zeichen lang ist.

@SFMITTEL(*Bereich*)

Berechnet den Standardfehler für den Mittelwert einer Stichprobe der Werte in *Bereich*.

@SIN(*x*)

Berechnet den Sinus eines Winkels *x*.

@SINHYP(*x*)

Berechnet den hyperbolischen Sinus eines Winkels *x*.

@SPALTEN(*Bereich*)

Zählt die Anzahl der Spalten in *Bereich*.

@SP(*Kapital;Zinssatz;Laufzeit;Periode*)

Berechnet den Zinsanteil einer periodischen Zahlung, bei der der Kapitalanteil in jeder Zahlungsperiode gleich ist.

@STANDARD(*x;[Mittel];[Stdabw];[Typ]*)

Berechnet die normale Distributionsfunktion für *x*.

@STDABW(*Argumentliste*)

Berechnet die Standardabweichung einer Grundgesamtheit von Werten in *Argumentliste*.

@STDABWP(*Argumentliste*)

Berechnet die Standardabweichung von Stichproben in *Argumentliste*.

@STÜCKZINS(*Valutierungstermin;Emission;Erst-Zins;Coupon;[Nennwert];[Zinstermine];[Basis]*)

Berechnet die aufgelaufenen Zinsen für Wertpapiere mit periodischen Zinszahlungen.

@STÜCKZINS2(*Valutierungstermin;Fälligkeit;Coupon;[Nennwert];[Zinstermine];[Emission];[Erst-Zins];[Art]*)

Berechnet die aufgelaufenen Zinsen für Wertpapiere mit periodischen Zinszahlungen anhand japanischer Konventionen.

@STUNDE(*Zeitseriennummer*)

Extrahiert die Stunde, ein Wert zwischen 0 (Mitternacht) und 23 (23:00), aus einer *Zeitseriennummer*.

@SUMME(*Argumentliste*)

Addiert die Werte in *Argumentliste*.

@SUMMENEGATIV(*Argumentliste*)

Addiert nur die negativen Werte in *Argumentliste*

@SUMMEPOSITIV(*Argumentliste*)

Addiert nur die positiven Werte in *Argumentliste*

@SUMPROD(*Argumentliste*)

Multipliziert die Werte in entsprechenden Zellen in einer *Argumentliste* von Bereichen und summiert dann die Produkte.

@SUMQUADRAT(*Argumentliste*)

Berechnet die Summe der Quadrate in *Argumentliste*.

@SUMXMY2(*Bereich1;Bereich2*)

Subtrahiert die Werte in *Bereich2* von den entsprechenden Zellen in *Bereich1*, potenziert die Differenz und addiert die Ergebnisse.

@SZENARIOINFO(*Option;Name;[Erstellt-von]*)

Liefert Informationen über eine Versionsgruppe.

@SZENARIOLETZT(*Dateiname*)

Liefert den Namen der zuletzt während der aktuellen 1-2-3 Sitzung in einer Arbeitsmappe angezeigten Versionsgruppe.

T

@TAG(*Datumseriennummer*)

Extrahiert den Tag des Monats, ein Wert zwischen 1 und 31, aus *Datumseriennummer*.

@TAGE(*Anfangsdatum;Enddatum;[Basis]*)

Berechnet die Anzahl von Tagen zwischen *Anfangsdatum* und *Enddatum*, unter Verwendung einer bestimmten *Zählbasis*.

@TAN(x)

Berechnet den Tangens eines Winkels x .

@TANHYP(x)

Berechnet den hyperbolischen Tangens eines Winkels x .

@THAIDATUMKURZ(Datumwert)

Konvertiert einen *Datumwert* in einen Thai-Datumsnamen im kurzen Format.

@THAIDATUMLANG(Datumwert)

Konvertiert einen *Datumwert* in einen Thai-Datumsnamen im langen Format.

@THAIERSETZEN(Original-Zeichenfolge;Startnummer;n;Neue-zeichenfolge)

Ersetzt n logische Thai-Zeichen in *Original-Zeichenfolge* mit der *Neuen-Zeichenfolge*, beginnend bei *Startnummer*. Wird zusammen mit Thai-Zeichenfolgen verwendet.

@THAIFINDEN(Such-Zeichenfolge;Zeichenfolge;Startnummer)

Berechnet die Spalte des ersten Vorkommens von *Suchfolge* in *Zeichenfolge*, beginnend mit dem Thai-Zeichen bei *Startnummer*. Wird zusammen mit Thai-Zeichenfolgen verwendet.

@THAILÄNGE(Zeichenfolge)

Zählt die Anzahl der logischen Thai-Zeichen in *Zeichenfolge*. Wird zusammen mit Thai-Zeichenfolgen verwendet.

@THAILINKS(Zeichenfolge;n)

Liefert die ersten n logischen Thai-Zeichen in *Zeichenfolge*. Wird zusammen mit Thai-Zeichenfolgen verwendet.

@THAIMITTE(Zeichenfolge;Startnummer;n)

Kopiert n logische Thai-Zeichen aus *Zeichenfolge*, beginnend mit dem Zeichen bei *Startnummer*. Wird zusammen mit Thai-Zeichenfolgen verwendet.

@THAIRECHTS(Zeichenfolge;n)

Ermittelt die n letzten logischen Thai-Zeichen in *Zeichenfolge*. Wird zusammen mit Thai-Zeichenfolgen verwendet.

@THAIWOCHENTAG(Datumwert)

Konvertiert einen *Datumwert* in einen Thai-Wochentag.

@THAIZAHLENFOLGE(Zahl)

Konvertiert eine *Zahl* in eine ausgeschriebene Thai-numerische Zeichenfolge.

@THAIZIFFERNFOLGE(Ziffernfolge)

Konvertiert eine arabische *Ziffernfolge* in eine Thai-numerische Zeichenfolge.

@TILGUNG(Kapital;Zinssatz;Laufzeit;Anfangsperiode;[Endperiode];[Typ];[Zukunft])

Berechnet den Kapitalanteil einer periodischen Zahlung eines Darlehens (*Kapital*) bei einem gegebenen *Zinssatz* für die angegebene Anzahl an Zahlungsperioden (*Laufzeit*).

@TTEST(Bereich1;Bereich2;[Typ];[Verteilungsende])

Führt einen Studentischen t -Test mit den Daten in *Bereich1* und *Bereich2* durch und gibt die zugehörige Wahrscheinlichkeit zurück.

@TVERTEILUNG(x;Freiheitsgrad;[Typ];[Verteilungsende])

Berechnet die Studentische t -Verteilung.

U

@UNGERADE(x)

Rundet einen Wert x zur nächsten ungeraden Ganzzahl auf.

V

@VAR(Argumentliste)

Berechnet die Abweichung einer Grundgesamtheit von Werten in *Argumentliste*.

@VARP(Argumentliste)

Berechnet die Abweichung von Stichproben in *Argumentliste*.

@VERGLEICH(Zellinhalt;Bereich;[Typ])

Ermittelt die relative Position der Zelle in *Bereich*, deren Inhalt mit *Zellinhalt* übereinstimmt.

@VERSIONAKTUELL(Bereich)

Liefert den Namen der aktuellen Version in *Bereich*.

@VERSIONINFO(Option;Versions-Bereich;Name;[Erstellt-von])

Liefert Informationen über eine Version.

@VERSIONSDATEN(*Option;Zelle;Versions-Bereich;Name;[Erstellt_von]*)

Ermittelt den Inhalt einer angegebenen Zelle in einer Version.

@VVERWEIS(*x;Bereich;Versatz*)

Findet den Inhalt einer Zelle in einer angegebenen Spalte einer vertikalen Verweistabelle.

W

@WAHL(*x;Argumentliste*)

Liefert das n-te Objekt aus *Argumentliste*, bei dem n gleich dem Wert von x ist.

@WAHR

Gibt den logischen Wert 0 (wahr) zurück.

@WENN(*Bedingung;x;y*)

Prüft eine *Bedingung* und gibt x zurück, wenn die *Bedingung* wahr ist, und y, wenn die *Bedingung* falsch ist.

@WERT(*Zeichenfolge*)

Setzt eine als *Zeichenfolge* eingegebene Zahl in den entsprechenden Wert um.

@WIEDERHOLEN(*Zeichenfolge;n*)

Dupliziert die *Zeichenfolge* n-mal.

@WOCHENTAG(*Datumseriennummer*)

Extrahiert den Tag des Monats, ein Wert zwischen 0 (Montag) und 6 (Sonntag), aus *Datumseriennummer*.

@WURZEL(*x*)

Berechnet die positive Quadratwurzel von x.

@WURZELPI(*x*)

Berechnet die Quadratwurzel von $x \cdot \pi$.

X

@XINDEX(*Bereich;Spaltentitel;Zeilentitel;[Arbeitsblattitel]*)

Ermittelt den Inhalt einer Zelle die durch *Spaltentitel*, *Zeilentitel* und *Arbeitsblattitel* bezeichnet ist.

@XINTZINS(*Schätzwert;Cash-flow;Datum*)

Berechnet die interne Ertragsrate für eine Reihe von Einnahmen und Ausgaben.

@XNETAKTWERT(*Zinssatz;Cash-flow;Datum*)

Berechnet den aktuellen Nettowert einer Reihe von Einnahmen und Ausgaben.

Y

Z

@ZAHLENFOLGE(*Zahl;typ*)

Wandelt eine *Zahl* in die ausgeschriebene japanische Zeichenfolge um und verwendet das in *Typ* angegebene Format.

@ZEICHEN(*x*)

Liefert das Zeichen des LMBCS-Zeichensatzes (Lotus Multibyte Character Set) zurück, das der Zahl x entspricht.

@ZEILEN(*Bereich*)

Zählt die Anzahl der Zeilen in *Bereich*.

@ZEIT(*Stunde;Minuten;Sekunden*)

Berechnet die Zeitseriennummer für die angegebene *Stunde*, die *Minuten* und die *Sekunden*.

@ZEITWERT(*Zeichenfolge*)

Berechnet die Zeitseriennummer für die Zeit in *Zeichenfolge*.

@ZELL(*Attribut;Adresse*)

Gibt Informationen über die erste Zelle in *Adresse* zurück

@ZELLZEIGER(*Attribut*)

Gibt Informationen über die aktuelle Zelle zurück.

@ZINS(*Zukwertvalue;Aktwert;Laufzeit*)

Berechnet den periodischen Zinssatz, bei dem eine Investition über die angegebene *Laufzeit* auf einen *Zukwert* anwächst.

@ZINSBETRAG(*Kapital;Zinssatz;Laufzeit;Anfangsperiode;[Endperiode];[Typ];[Zukwert]*)

Berechnet den kumulativen Zinsanteil einer periodischen Zahlung für eine angegebene Anzahl Zahlungsperioden (*Laufzeit*).

@ZTEST(*Bereich1;Mittel1;Stdabw1;[Verteilungsende];[Bereich2];[Mittel2];[Stdabw2]*)

Führt einen z-Test mit einer oder zwei Populationen durch und gibt die zugehörige Wahrscheinlichkeit zurück.

@ZUFALLSZAHL

Erzeugt einen Zufallswert zwischen 0 und 1 mit bis zu 15 signifikanten Stellen. @ZUFALLSZAHL erzeugt bei jeder Neuberechnung einen neuen Zufallswert.

@ZUKWERT(*Zahlungen;Zinssatz;Laufzeit*)

Berechnet den zukünftigen Wert einer Investition, basierend auf einer Reihe gleicher *Zahlungen* bei einem periodischen *Zinssatz* über die Anzahl der Zahlungsperioden in *Laufzeit* bei nachschüssiger Rente.

@ZUKWERT2(*Zahlungen;Zinssatz;Laufzeit*)

Berechnet den zukünftigen Wert einer Investition, basierend auf einer Reihe gleicher *Zahlungen* bei einem periodischen *Zinssatz* über die Anzahl der Zahlungsperioden in *Laufzeit* bei vorschüssiger Rente.

@ZWERT(*Zahlungen;Zinssatz;Laufzeit;[Typ];[Aktwert]*)

Berechnet den zukünftigen Wert einer Kapitalanlage mit einem *aktuellen Wert* bei vorschüssiger oder nachschüssiger Rente.

@ZWERTBETRAG(*Kapital;Zinssatz;Laufzeit;[Zinstermine]*)

Berechnet den zukünftigen Wert eines Pauschalbetrages, der zu einem bestimmten Satz für eine bestimmte Anzahl von Perioden investiert wird.

@ZWISCHENSUMME(*Argumentliste*)

Addiert die Werte in *Argumentliste*. Verwenden Sie @ZWISCHENSUMME um anzugeben, welche Zellen @ENDSUMME addieren soll.

@ABS

@ABS(x) berechnet den absoluten (positiven) Wert von x.

Argumente

x ist ein beliebiger Wert.

Hinweise

Benutzen Sie -@ABS, wenn das Ergebnis der @Funktion negativ sein soll.

Beispiele

@ABS(A5) = 25, wenn Zelle A5 den Wert 25, -25 oder eine Formel enthält, die 25 oder -25 ergibt.

-@ABS(A5) = -25, wenn Zelle A5 den Wert 25, -25 oder eine Formel enthält, die 25 oder -25 ergibt.

@ABS(ANFANG-ENDE) = 5,6, wenn ANFANG und ENDE eine beliebige Kombination aus positiven oder negativen Zahlen enthalten, die sich um 5,6 oder -5,6 unterscheiden.

@STÜCKZINS

@STÜCKZINS(*Valutierungstermin*; *Emission*; *Erstzins*; *Coupon*; [*Nennwert*]; [*Zinstermine*]; [*Zinstage*]) berechnet die aufgelaufenen Zinsen für Anleihen mit periodischen Zinszahlungen. @STÜCKZINS unterstützt kurz-, mittel- und langfristige Zahlungsperioden.

Argumente

Valutierungstermin ist eine Datumseriennummer und steht für das Abrechnungsdatum einer Anleihe. Ist *Valutierungstermin* kleiner als *Emission*, so ergibt @STÜCKZINS den Wert FEHLER.

Emission ist eine Datumseriennummer und steht für das Emissionsdatum einer Anleihe

Erstzins ist eine Datumseriennummer und steht für das erste Zinsdatum der Anleihe. Wenn *Erstzins* kleiner oder gleich *Emission* ist, ergibt @STÜCKZINS den Wert FEHLER.

Coupon ist der jährliche Anleihezins. *Coupon* ist ein beliebiger Wert oder 0.

Nennwert ist ein wahlweises Argument, das den Nennwert der Anleihe angibt, d.h. die Kapitalsumme, die beim Fälligkeitstermin ausbezahlt wird. *Nennwert* ist ein positiver Wert. Wenn Sie das *Nennwert*-Argument nicht angeben, benutzt 1-2-3 den Vorgabewert 100.

Zinstermine ist ein wahlweises Argument, das angibt, wie oft die Zinsen jährlich bezahlt werden. *Zinstermin* kann folgende Werte annehmen:

| Zinstermine | Häufigkeit der Zinszahlungen |
|--------------------|--|
| 1 | Jährlich |
| 2 | Halbjährlich; Vorgabe, wenn Sie das Argument nicht angeben |
| 4 | Vierteljährlich |
| 12 | Monatlich |

Zinstage ist ein wahlweises Argument, mit dem angegeben wird, auf welcher Basis die Zinstage berechnet werden. *Zinstage* entspricht einem Wert aus der folgenden Tabelle:

| Zinstage | Berechnungsgrundlage |
|-----------------|--|
| 0 | 30/360; Vorgabe, wenn Sie das Argument nicht angeben |
| 1 | Anzahl/Anzahl |
| 2 | Anzahl/360 |
| 3 | Anzahl/365 |
| 4 | Europäisch 30/360 |

Sie können ein wahlweises Argument nicht ohne die vorhergehenden Argumente angeben.

Beispiele

Eine Anleihe hat als Valutierungstermin den 1. Juli 1993 und als Emissionsdatum den 1. Dezember 1992. Das Datum des Erstzinses fällt auf den 1. Juni 1993. Der halbjährliche Coupon beläuft sich auf 5,50%. Die Anleihe hat einen Nennwert von 100 DM und wird auf einer Basis von 30/360 Zinstagen abgerechnet.

Um den Stückzins der Anleihe zu ermitteln, benutzen Sie folgende @Funktion:

@STÜCKZINS(@DATUM(93;7;1);@DATUM(92;12;1);@DATUM(93;6;1);0,055;100;2;0) = 0,458333

Ähnliche @Funktionen

@STÜCKZINS2 berechnet den Stückzins für Wertpapiere mit periodischen Zinszahlungen unter Verwendung japanischer Konventionen.

@KURS berechnet den Kurs pro 100 DM Nennwert für eine Anleihe. @RENDITE berechnet die Rendite von Anleihen, für die periodische Zinszahlungen vorgenommen werden. @LAUFZEIT berechnet die jährliche Laufzeit und @MLAUFZEIT die modifizierte jährliche Laufzeit von Anleihen, für die in regelmäßigen Abständen Zinsen bezahlt werden.

@ARCCOS

@ARCCOS(x) berechnet den Arkuskosinus (inversen Kosinus) mit dem Kosinus x eines Winkels. Das Ergebnis von **@ARCCOS** ist ein Winkel in Radian von 0 bis π . Dies entspricht einem Winkel zwischen 0 und 180 Grad.

Argumente

x ist der Kosinus eines Winkels und ist ein Wert von -1 bis 1.

Beispiele

Der Kosinus des Winkels z ist 0,5. Um die Größe des Winkels z zu berechnen, verwenden Sie **@ARCCOS**(0,5). Das Ergebnis ist 1,0472 Radian. Um diesen Wert in Grad umzuwandeln, benutzen Sie **@RADINGRAD**(1,0472). Das Ergebnis ist 60 Grad.

Ähnliche @Funktionen

@COS berechnet den Kosinus eines Winkels. **@ARCOSH** berechnet den hyperbolischen Arkuskosinus eines Wertes.

@ARCOSH

@ARCOSH(x) berechnet den hyperbolischen Arkuskosinus (inversen hyperbolischen Kosinus), indem es den hyperbolischen Kosinus x eines Winkels benutzt.

Argumente

x ist der hyperbolische Kosinus eines Winkels und ein Wert größer als oder gleich 1.

Examples

@ARCOSH(2) = 1,316958

Ähnliche @Funktionen

@ARCCOS berechnet den Arkuskosinus eines Wertes. **@COSH** berechnet den hyperbolischen Kosinus eines Winkels.

@ARCCOT

@ARCCOT(x) berechnet den Arkuskotangens (inversen Kotangens) mit dem Kotangens x eines Winkels. Das Ergebnis von **@ARCCOT** ist ein Winkel in Radian von 0 bis π . Dies entspricht einem Winkel zwischen 0 und 180 Grad.

Argumente

x ist der Kotangens eines Winkels und entspricht einem Wert von $-2^{32} \cdot \pi$ bis $2^{32} \cdot \pi$.

Beispiele

Der Kotangens des Winkels z ist 1,732051. Um die Größe des Winkels z zu bestimmen, verwenden Sie **@ARCCOT**(1,732051). Dies ergibt 0,523599 Radian. Um diesen Wert in Grad umzuwandeln, verwenden Sie **@RADINGRAD**(0,523599). Das Ergebnis entspricht 30 Grad.

Ähnliche @Funktionen

@COT berechnet den Kotangens eines Winkels. **@ATAN** berechnet den Arkustangens eines Wertes.

@ARCOTH

@ARCOTH(x) berechnet den hyperbolischen Arkuskotangens (hyperbolischen inversen Kotangens) mit dem hyperbolischen Kotangens x eines Winkels.

Argumente

x ist der hyperbolische Kotangens eines Winkels und entspricht einem Wert von kleiner als -1 oder größer als 1.

Beispiele

@ACOTH(2) = 0,549306

Ähnliche @Funktionen

@ARCCOT berechnet den Arkuskotangens eines Wertes. @COTH berechnet den hyperbolischen Kotangens eines Winkels.

@ARCCOSEC

@ARCCOSEC(x) berechnet den Arkuskosekans (inversen Kosekans) mit dem Kosekans x eines Winkels. Das Ergebnis von **@ARCCOSEC** ist ein Winkel in Radiant von $-\pi/2$ bis $\pi/2$. Dies entspricht einem Winkel zwischen -90 und 90 Grad.

Argumente

x ist der Kosekans eines Winkels und entspricht einem Wert größer als oder gleich 1 bzw. kleiner als oder gleich -1.

Beispiele

Der Kosekans des Winkels z ist 1,743447. Um die Größe des Winkels z zu bestimmen, verwenden Sie **@ARCCOSEC**(1,743447). Dies ergibt 0,610865 Radian. Um diesen Wert in Grad umzuwandeln, verwenden Sie **@RADINGRAD**(0,610865). Das Ergebnis entspricht 35 Grad.

Ähnliche @Funktionen

@ARCOSECH berechnet den hyperbolischen Arkuskosekans eines Wertes. **@COSEC** berechnet den Kosekans eines Winkels.

@ARCOSECH

@ARCOSECH(x) berechnet den hyperbolischen Arkuskosekans (hyperbolischen inversen Kosekans) mit dem hyperbolischen Kosekans x eines Winkels.

Argumente

x ist der hyperbolische Kosekans eines Winkels und entspricht einem anderen Wert als 0.

Beispiele

@ACSCH(1.54) = 0,61068

Ähnliche @Funktionen

@ARCCOSEC berechnet den Arkuskosekans eines Wertes. @COSECH berechnet den hyperbolischen Kosekans eines Winkels.

@ARCSEC

@ARCSEC(x) berechnet den Arkussekans (inversen Sekans) mit dem Sekans x eines Winkels. Das Ergebnis von **@ARCSEC** ist ein Winkel in Radian von 0 bis π . Dies entspricht einem Winkel zwischen 0 und 180 Grad.

Argumente

x ist der Sekans eines Winkels und entspricht einem Wert kleiner als oder gleich -1 oder größer als oder gleich 1.

Beispiele

In einem rechtwinkligen Dreieck ist der Sekans des Winkels z 2. Um die Größe des Winkels z zu bestimmen, verwenden Sie **@ARCSEC(2)**. Dies ergibt 1,047198 Radian. Um diesen Wert in Grad umzuwandeln, verwenden Sie **@RADINGRAD(1,047198)**. Das Ergebnis ist 60 Grad.

Ähnliche @Funktionen

@SEC berechnet den Sekans eines Winkels. **@ARCSECH** berechnet den hyperbolischen Arkussekans eines Wertes.

@ARSECH

@ARSECH(x) berechnet den hyperbolischen Arkussekans (hyperbolischen inversen Sekans) mit dem hyperbolischen Sekans x eines Winkels.

Argumente

x ist der hyperbolische Sekans eines Winkels und entspricht einem Wert größer als 0 und kleiner als oder gleich 1.

Beispiele

@ASECH(0,5) = 1,316958

Ähnliche @Funktionen

@ARCSEC berechnet den Arkussekans eines Winkels. **@SECH** berechnet den hyperbolischen Sekans eines Winkels.

@ARCSIN

@ARCSIN(x) berechnet den Arkussinus (inversen Sinus) mit dem Sinus x eines Winkels. Das Ergebnis von @ARCSIN ist ein Winkel in Radiant von $-\pi/2$ bis $\pi/2$. Dies entspricht einem Winkel zwischen -90 und 90 Grad.

Argumente

x ist der Sinus eines Winkels und entspricht einem Wert von -1 bis 1.

Beispiele

Der Sinus des Winkels z ist 0,66. Um die Größe des Winkels z zu bestimmen, verwenden Sie @ARCSIN(0,66). Dies ergibt 0,72082 Radian. Um diesen Wert in Grad umzuwandeln, verwenden Sie @RADINGRAD(0,72082). Das Ergebnis ist 41,3 Grad.

Ähnliche @Funktionen

@SIN berechnet den Sinus eines Winkels. @ARSINH berechnet den hyperbolischen Arkussinus eines Wertes.

@ARSINH

@ARSINH(x) berechnet den hyperbolischen Arkussinus (hyperbolischen inversen Sinus) mit dem hyperbolischen Sinus x eines Winkels.

Argumente

x ist der hyperbolische Sinus eines Winkels und entspricht einem beliebigen Wert.

Beispiele

@ARSINH(2) = 1,443635

Ähnliche @Funktionen

@SINH berechnen den hyperbolischen Sinus eines Winkels. @ARCSIN berechnet den Arkussinus eines Wertes.

@@

Mit @@(*Adresse*) erhalten Sie indirekt den Inhalt der Zelle, die in *Adresse* angegeben wird.

Argumente

Adresse ist die Adresse oder der Name einer Zelle, die eine Zelladresse oder einen Namen enthält, bzw. eine Formel, die die Adresse oder den Namen einer Zelle zurückgibt. *Adresse* zeigt auf eine andere Zelle, deren Inhalt @@ in der Zelle anzeigt, in der @@ steht. Ist *Adresse* keine gültige Zelladresse bzw. kein gültiger Bereichsname oder handelt es sich um einen aus mehreren Zellen bestehenden Bereich, so ergibt @@ FEHLER.

Hinweise

@ @ wird besonders beim Aufbau bedingter Formeln benutzt, da sie durch ihren indirekten Bezug auf wechselnde Inhalte in der gewünschten Zelle reagiert. Die Formel

@ @WENN(D2="J";"D8";@WENN(D2="N";"D9";@FEHLER))

in Zelle A10 und die Formel @@(A10) in Zelle E2 geben z.B. den Inhalt der Zelle D8 oder D9 bzw. FEHLER in E2 zurück, je nachdem, ob D2 J oder N oder eine andere Angabe enthält.

Bezieht sich *Adresse* auf eine Zelle, die eine Formel enthält, so müssen Sie F9 (KALK) drücken, um die @@ Formel nach der automatischen Neuberechnung zu aktualisieren. Wenn Sie F9 (KALK) nicht drücken, ergibt die @@ Formel 0.

Beispiele

@@

Ähnliche @Funktionen

@HVERWEIS und @VVERWEIS geben den Inhalt einer bestimmten Zelle in einer horizontalen oder vertikalen Verweistabelle zurück. @WAHL gibt einen Wert oder ein Label aus *Argument-Liste* zurück, und @INDEX gibt den Inhalt einer Zelle zurück, die bei dem Schnittpunkt einer bestimmten Spalte mit einer bestimmten Zeile (in dem angegebenen Arbeitsblatt) steht.

@ARCTAN

@ARCTAN(x) berechnet den Arkustangens (inversen Tangens) mit dem Tangens x eines Winkels. Das Ergebnis von **@ARCTAN** ist ein Winkel in Radian von $-\pi/2$ bis $\pi/2$. Dies entspricht einem Winkel zwischen -90 und 90 Grad.

Argumente

x ist der Tangens eines Winkels und entspricht einem beliebigen Wert.

Beispiele

Der Tangens des Winkels z ist $2/1$ oder 2 . Um die Größe des Winkels z zu bestimmen, verwenden Sie **@ARCTAN(2)**. Dies ergibt $1,10715$ Radian. Um diesen Wert in Grad umzuwandeln, verwenden Sie **@RADINGRAD(1,10715)**. Das Ergebnis entspricht $63,4$ Grad.

Ähnliche @Funktionen

@ARTANH berechnet den hyperbolischen Arkustangens eines Wertes. **@ARCTAN2** berechnet die Größe eines Winkels, dessen Tangens y/x ist. **@TAN** berechnet den Tangens eines Winkels.

@ARCTAN2

@ARCTAN2(x;y) berechnet den Arkustangens unter Verwendung des Tangens y/x eines Winkels. Das Ergebnis von **@ARCTAN2** ist ein Winkel in Radiant von $-p$ bis p . Dies entspricht einem Winkel zwischen -180 und 180 Grad, je nach Vorzeichen von x und y (siehe Liste unten).

Argumente

x und y sind Werte. Wenn y gleich 0 ist, ergibt **@ARCTAN2** den Wert 0 ; wenn x und y gleich 0 sind, ergibt **@ARCTAN2** den Wert FEHLER.

Hinweise

In der folgenden Liste finden Sie die Wertbereiche für **@ARCTAN2**.

- Wenn x positiv und y positiv ist, kann das Ergebnis von 0 bis $p/2$ gehen (Quadrant I).
- Wenn x negativ und y positiv ist, kann das Ergebnis von $p/2$ bis p gehen (Quadrant II).
- Wenn x negativ und y negativ ist, kann das Ergebnis von $-p$ bis $-p/2$ gehen (Quadrant III).
- Wenn x positiv und y negativ ist, kann das Ergebnis von $-p/2$ bis 0 gehen (Quadrant IV).

Wenn x und y beide positiv sind (Quadrant I) und wenn x positiv und y negativ ist (Quadrant IV), sind die Ergebnisse dieselben wie bei **@ARCTAN**.

Beispiele

In einem rechtwinkligen Dreieck haben die beiden Seiten, die den rechten Winkel bilden, den Wert 1 und 2 . Um die Größe des größeren der beiden spitzen Winkel zu bestimmen, verwenden Sie **@ARCTAN2(1;2)**. Dies ergibt $1,10715$ Radian. Um diesen Wert in Grad umzuwandeln, verwenden Sie **@RADINGRAD(1,10715)**. Das Ergebnis entspricht $63,4$ Grad.

Ähnliche @Funktionen

@ARCTAN berechnet den Arkustangens mit dem Tangens x eines Winkels. **@TAN** berechnet den Tangens eines Winkels.

@ARTANH

@ARTANH(x) berechnet den hyperbolischen Arkustangens (hyperbolischen inversen Tangens) mit dem hyperbolischen Tangens x. Das Ergebnis von @ARTANH ist ein Winkel in Radian.

Argumente

x ist der hyperbolische Tangens eines Winkels und entspricht einem Wert zwischen -1 und 1.

Beispiele

@ATANH(0,544736) = 0,610865

Ähnliche @Funktionen

@ARCTAN berechnet den Arkustangens eines Wertes. @ARCTAN2 berechnet die Größe eines Winkels, dessen Tangens y/x ist. @TANH berechnet den hyperbolischen Tangens eines Winkels.

@MITTELABW

@MITTELABW(*Argument-Liste*) berechnet den Mittelwert der absoluten Abweichungen der Werte in *Argument-Liste*.

Argumente

Argument-Liste kann jede der folgenden Angaben in einer beliebigen Kombination enthalten: Zahlen, numerische Formeln und Adressen oder Namen von Bereichen, die Zahlen oder numerische Formeln enthalten. Trennen Sie die Elemente in der *Argument-Liste* durch Argument-Trennzeichen.

Siehe auch Argumente für statistische @Funktionen.

Beispiele

@MITTELABW

Ähnliche @Funktionen

@STDABW und @PURSTDABW berechnen die Standardabweichung der Werte in einer Argument-Liste.

@ABWQUADRAT berechnet die Summe der quadratischen Abweichung der Werte in einer Argument-Liste.

@MITTELWERT

@PURMITTELWERT

@MITTELWERT(*Argument-Liste*) berechnet den Mittelwert einer *Liste* mit Werten.

@PURMITTELWERT(*Argument-Liste*) berechnet den Mittelwert einer *Liste* mit Werten, wobei alle Zellen ignoriert werden, die Labels enthalten.

Argumente

Argument-Liste kann jede der folgenden Angaben in einer beliebigen Kombination enthalten: Zahlen, numerische Formeln und Adressen oder Namen von Bereichen, die Zahlen oder numerische Formeln enthalten. Trennen Sie die Elemente in *Argument-Liste* durch Argument-Trennzeichen.

Siehe auch Argumente für statistische @Funktionen.

Beispiele

@MITTELWERT(A2..A5) = 252,75, wobei A2..A5 die Werte 160, 227, 397 und 227 enthält.

@MITTELWERT(A1..A5) = 202,20, wobei A1..A5 die Werte 160, 227, 397 und 227 und das Label Januar enthält. (@MITTELWERT zählt das Label als null und benutzt es in der Berechnung.)

@PURMITTELWERT(A1..A5) = 252,75, da @PURMITTELWERT das Label Januar ignoriert.

Ähnliche @Funktionen

@DMITTELWERT berechnet den Mittelwert der Werte in einem Feld einer Datenbanktabelle, die den von Ihnen angegebenen Kriterien entsprechen. @GEOMITTEL berechnet das geometrische Mittel und @HARMITTEL berechnet das harmonische Mittel der Werte in einer Liste. @MEDIAN berechnet den Medianwert in einer Liste mit Werten.

@BESSELI

@BESSELJ

@BESSELK

@BESSELY

@BESSELI($x;n$) berechnet die geänderte Bessel-Funktion ganzzahliger Ordnung $I_n(x)$.

@BESSELJ($x;n$) berechnet die Bessel-Funktion ganzzahliger Ordnung $J_n(x)$.

@BESSELK($x;n$) berechnet die geänderte Bessel-Funktion ganzzahliger Ordnung $K_n(x)$.

@BESSELY($x;n$) berechnet die Bessel-Funktion ganzzahliger Ordnung $Y_n(x)$, auch als Neumann-Funktion bekannt.

Argumente

x ist der Wert, mit dem die Funktion ausgewertet wird. Hier kann es sich um einen beliebigen Wert handeln.

n ist die Ordnung der Funktion und ist eine positive Ganzzahl. Für @BESSELI und @BESSELJ kann n auch 0 sein.

Hinweise

@BESSELI, @BESSELJ, @BESSELK und @BESSELY berechnen die entsprechenden Funktionen näherungsweise innerhalb des Bereichs $\pm 5 \cdot 10^{-8}$.

Bessel-Funktionen werden häufig in Aufgaben mit zylindrischer Symmetrie benutzt, in Verbindung mit Wellenausbreitung, Flüssigkeitsbewegung, Elastizität und Diffusion.

Beispiele

@BESSELI(2;2) = 0,688948

@BESSELJ(1;0) = 0,765198

@BESSELK(3;0) = 0,03474

@BESSELY(1;1) = -0,781213

@BETA

@BETA($x;w$) berechnet die Beta-Funktion.

Argumente

x und w können beliebige Werte sein.

Hinweise

Das Ergebnis von @BETA ist innerhalb von mindestens sechs signifikanten Ziffern genau.

Beispiele

@BETA(0,5;0,5) = 3,141593

Ähnliche @Funktionen

@BETA! berechnet die unvollständige Beta-Funktion. @GAMMA berechnet die Gamma-Funktion.

@BETAI

@BETAI($a;b;x$) berechnet die unvollständige Beta-Funktion.

Argumente

a und b können beliebige Werte darstellen.

x ist ein Wert von 0 bis 1.

Hinweise

Das Ergebnis von @BETAI ist innerhalb von mindestens sechs signifikanten Ziffern genau.

Beispiele

@BETAI(5;0,5;0,668271) = 0,050012

Ähnliche @Funktionen

@BETA berechnet die Beta-Funktion.

@BINOMIAL

@BINOMIAL(*Versuche*; *Ergebnisse*; *Wahrscheinlichkeit*; [*Typ*]) berechnet die binomiale Wahrscheinlichkeits-Massenfunktion oder die kumulative Binomialverteilung.

Argumente

Versuche ist die Anzahl von unabhängigen Versuchen. *Versuche* ist eine beliebige positive Ganzzahl.

Ergebnisse ist die Anzahl von Ergebnissen in *Versuche* und stellt eine positive Ganzzahl oder 0 dar und muß kleiner als oder gleich den *Versuchen* sein.

Wenn es sich bei *Versuche* und *Ergebnisse* nicht um Ganzzahlen handelt, schneidet 1-2-3 diese auf Ganzzahlen ab.

Wahrscheinlichkeit ist die Erfolgswahrscheinlichkeit bei jedem Versuch und entspricht einem Wert von 0 bis 1.

Typ ist ein wahlweises Argument, mit dem angegeben wird, ob 1-2-3 die Wahrscheinlichkeits-Massenfunktion oder die kumulative Binomialverteilung berechnet.

| Typ | 1-2-3 berechnet |
|------------|--|
| 0 | Die Wahrscheinlichkeit von genauen <i>Ergebnissen</i> Anzahl von Ergebnissen; Vorgabe, wenn Sie das Argument weglassen |
| 1 | Die Wahrscheinlichkeit von maximalen <i>Ergebnissen</i> Anzahl von Ergebnissen |
| 2 | Die Wahrscheinlichkeit von mindesten <i>Ergebnissen</i> Anzahl von Ergebnissen |

Hinweise

@BINOMIAL berechnet die kumulative Binomialverteilung näherungsweise innerhalb des Bereichs $\pm 3 \cdot 10^{-7}$.

Beispiele

Sie wählen zufällig zehn Colatrinker, die an einem blinden Geschmackstest teilnehmen sollen. Sie geben jedem Teilnehmer ein Glas Cola A und ein Glas Cola B. Die Gläser sehen genau gleich aus, mit Ausnahme eines Codes auf der Unterseite zur Kennzeichnung der Cola-Art. Wenn davon ausgegangen wird, daß unter den Colatrinkern keine Tendenz besteht, eine Colamarke gegenüber einer anderen zu bevorzugen, so ist die Wahrscheinlichkeit, daß einem Testteilnehmer Cola A besser schmeckt, gleich 50%.

Um die Wahrscheinlichkeit zu berechnen, daß genau 7 von 10 Testteilnehmern Cola A bevorzugen, benutzen Sie folgende @Funktion:

@BINOMIAL(10;7;0,5) = 0,117188 oder 11,72%.

Um die Wahrscheinlichkeit zu berechnen, daß mindestens 7 von 10 Testteilnehmern Cola A bevorzugen, benutzen Sie folgende @Funktion:

@BINOMIAL(10;7;0,5;2) = 0,171875 oder 17,19%.

Ähnliche @Funktionen

@KRITBINOMIAL berechnet den größten Wert, für den die kumulative Binomialverteilung kleiner als oder gleich einem bestimmten Kriterium ist. @KOMBIN berechnet die Anzahl von Kombinationen für eine bestimmte Anzahl von Werten. @PERMUT berechnet die Anzahl von Permutationen für eine Liste mit Werten.

@ZELLE, @ZELLZEIGER

@ZELLE(*Attribut;Ort*) gibt Informationen über die erste Zelle in *Ort* zurück.

@ZELLZEIGER(*Attribut*) gibt Informationen über die aktuelle Zelle zurück.

Argumente

Ort ist die Adresse oder der Name eines Bereichs.

Attribut ist eines der unten aufgeführten Elemente in " " (Anführungszeichen) bzw. die Adresse oder der Name einer Zelle, die eines der Elemente enthält.

| Attribut | Ergebnis |
|-----------------|--|
| Adresse | Die absolute Zelladresse in abgekürzter Form (nur Spaltenbuchstabe und Zeilennummer). |
| Blatt | Der Arbeitsblattbuchstabe als Wert von 1 bis 256 (1 für Blatt A, 2 für Blatt B usw.). |
| Blattname | Der Arbeitsblattname, oder, falls das Arbeitsblatt keinen Namen hat, der Arbeitsblattbuchstabe. |
| Breite | Die Spaltenbreite. |
| Dateidatum | Ein Wert, mit dem das Datum und die Zeit der letzten Speicherung der Datei angegeben wird, in der die Zelle steht. Dies umfaßt sowohl eine <u>Datumseriennummer</u> (ganzzahliger Teil) als auch eine <u>Zeitseriennummer</u> (Dezimalteil). |
| Dateiname | Der Name der Arbeitsmappe, die die Zelle enthält, einschließlich Pfad. |
| Datentyp | Der Datentyp in der Zelle: b, wenn die Zelle leer ist w, wenn die Zelle eine Zahl oder numerische Formel enthält l, wenn die Zelle ein Label oder eine Textformel enthält f, wenn die Zelle den Wert <u>FEHLER</u> enthält n, wenn die Zelle den Wert <u>NV</u> enthält |
| Drehung | Ein Drehwinkel für gedrehten Text in Form einer Ganzzahl von 0 bis 90 Grad. |
| Farbe | 1, wenn negative Werte in Farbe angezeigt werden. 0, wenn negative Werte nicht in Farbe angezeigt werden. |
| Fett | 1, wenn die Zelle fett formatiert ist. 0, wenn die Zelle nicht fett formatiert ist. |
| Format | Das Zellformat: W0 bis W15 bei Währung, 0 bis 15 Dezimalstellen F0 bis F15 bei Festem Format, 0 bis 15 Dezimalstellen S bei Standard, einem Label oder einer leeren Zelle P0 bis P15 bei Prozent, 0 bis 15 Dezimalstellen |

| | |
|------------------|--|
| | <p>E0 bis E15 bei Exponentialformat, 0 bis 15 Dezimalstellen</p> <p>.0 bis .15 bei . (Punkt), 0 bis 15 Dezimalstellen</p> <p>+ bei +/- Format</p> <p>D1 bei Datumsformat 31-Dez-92</p> <p>D2 bei Datumsformat 31-Dez</p> <p>D3 bei Datumsformat Dez-92</p> <p>D4 bei Datumsformat 12/31/92</p> <p>D5 bei Datumsformat 12/31</p> <p>U1 bei Zeitformat 11:59:59 AM</p> <p>U2 bei Zeitformat 11:59 AM</p> <p>U3 bei Zeitformat 23:59:59</p> <p>U4 bei Zeitformat 59:59</p> <p>T bei Text</p> <p>V bei Verborgenenem Format</p> <p>L bei Label-Format</p> <p>A bei Automatischem Format</p> <p>-- wenn Farbe bei negativen Werten</p> <p>() bei Klammern</p> |
| Formeltyp | <p>Der Formeltyp in der Zelle.</p> <p>b, wenn die Zelle leer ist</p> <p>w, wenn die Zelle eine Zahl enthält</p> <p>l, wenn die Zelle ein Label enthält</p> <p>fw, wenn die Zelle eine numerische Formel enthält</p> <p>ft, wenn die Zelle eine Textformel enthält</p> <p>ff, wenn die Zelle eine Formel enthält, deren Auswertung FEHLER ergibt</p> <p>fn, wenn die Zelle eine Formel enthält, deren Auswertung NV ergibt</p> |
| H_Ausricht | <p>Die Horizontalausrichtung der Daten in der Zelle:</p> <p>0 Standard (Labels, linksbündig ausgerichtet, Werte rechtsbündig)</p> <p>1 Linksbündig</p> <p>2 Zentriert</p> <p>3 Rechtsbündig</p> <p>4 Gleiche Abstände</p> |
| Hintergrund | <p>Die Hintergrundfarbe in Form einer Ganzzahl von 0 bis 239, mit der eine Farbe aus der Farbpalette angegeben wird.</p> |
| Hoch-/Querformat | <p>Das Hoch- bzw. Querformat für gedrehten Text in Form einer Ganzzahl von 0 bis 4, mit der eine Ausrichtung angegeben wird.</p> |
| Höhe | <p>Die Zeilenhöhe in Punkten.</p> |
| Inhalt | <p>Der Inhalt der Zelle.</p> |
| Klammern | <p>1, wenn die Zelle für Klammern formatiert ist.</p> <p>0, wenn die Zelle nicht für Klammern formatiert ist.</p> |
| Koord | <p>Die absolute Zelladresse in ausgeschriebener Form (Arbeitsblattbuchstabe, Spaltenbuchstabe und Zeilennummer).</p> |

| | |
|-----------------|--|
| Kursiv | 1, wenn die Zelle kursiv formatiert ist. 0, wenn die Zelle nicht kursiv formatiert ist. |
| Linkerrand | Eine Ganzzahl von 0 (kein Rand) bis 8, mit der ein Linienstil angegeben wird. |
| L_Randfarbe | Eine Ganzzahl von 0 bis 239, mit der eine Linienfarbe angegeben wird. |
| Muster | Eine Ganzzahl von 0 bis 63, mit der ein Muster angegeben wird. |
| Musterfarbe | Eine Musterfarbe in Form einer Ganzzahl von 0 bis 239, mit der eine Farbe aus der Farbpalette angegeben wird. |
| Obereinrahmung | Eine Ganzzahl von 0 (kein Rand) bis 8, mit der ein Linienstil angegeben wird. |
| Präfix | ' für ein linksbündiges Label " für ein rechtsbündiges Label ^ für ein zentriertes Label \ für ein Tabellenlabel für ein nicht-druckendes Label Leer (kein Labelpräfix), wenn die Zelle leer ist, oder einen Wert enthält |
| O_Randfarbe | Eine Ganzzahl von 0 bis 239, mit der eine Linienfarbe angegeben wird. |
| R_Randfarbe | Eine Ganzzahl von 0 bis 239, mit der eine Linienfarbe angegeben wird. |
| Rechteinrahmung | Eine Ganzzahl von 0 bis 15, mit der ein Linienstil angegeben wird. |
| Schriftart | Die Schriftart der Daten in der Zelle. |
| Schriftgröße | Die Schriftgröße der Daten in der Zelle. |
| Schutz | 1, wenn die Zelle geschützt ist. 0, wenn die Zelle nicht geschützt ist. |
| Spalte | Der Spaltenbuchstabe als Wert von 1 bis 256 (1 für Spalte A, 2 für Spalte B usw.). |
| Textfarbe | Die Farbe der Daten in der Zelle in Form einer Ganzzahl von 0 bis 239, mit der eine Farbe aus der Farbpalette angegeben wird. |
| Typ | Der Datentyp in der Zelle: b, wenn die Zelle leer ist w, wenn die Zelle einen numerischen Wert oder eine <u>numerische Formel</u> bzw. eine <u>Textformel</u> enthält l, wenn die Zelle ein Label enthält |
| Überspalten | 1, wenn die Daten in der Zelle über Spalten ausgerichtet sind. 0, wenn die Daten in der Zelle nicht über Spalten ausgerichtet sind. |
| Umbruch | 1, wenn die Daten in der Zelle umbrochen werden 0, wenn die Daten in der Zelle nicht umbrochen werden |

| | |
|-----------------|---|
| Untereinrahmung | Eine Ganzzahl von 0 (kein Rand) bis 8, mit der ein Linienstil angegeben wird. |
| Unterstreichen | Eine Ganzzahl von 0 (keine Unterstreichung) bis 3, mit der ein Unterstreichungsstil angegeben wird. |
| V_Ausricht | Die Vertikalausrichtung der Daten in der Zelle: 0 Unten 1 Zentriert 2 Oben |
| U_Randfarbe | Eine Ganzzahl von 0 bis 239, mit der eine Linienfarbe angegeben wird. |
| Zeile | Die Zeilennummer, von 1 bis 8192. |

Hinweise

Drücken Sie F9 (KALK), um die Daten neu zu berechnen, bevor Sie @ZELLE oder @ZELLZEIGER benutzen, um zu gewährleisten, daß die Ergebnisse richtig sind.

@ZELLE und @ZELLZEIGER werden insbesondere in Makros und in Verbindung mit @WENN benutzt. Mit @ZELLE können Sie die Eingabe während eines Makros überprüfen und damit bestimmte Eingaben ausschließen. Außerdem können Sie mit dieser @Funktion die Makroausführung steuern, indem Sie Sprünge zu Unterprogrammen von der Eingabe des Benutzers abhängig machen. Außerdem können Sie mit @ZELLE die Zellattribute je nach Benutzereingabe in automatisierten Anwendungen ändern.

Mit @ZELLZEIGER ermitteln Sie die aktuelle Position des Zellzeigers oder werten eine Formel aufgrund des Inhalts der aktuellen Zelle aus. Danach können Sie die Verarbeitung je nach Zellinhalt oder Zelltyp steuern.

Beispiele

In dem folgenden Beispiel wird @ZELLE mit @WENN und @FEHLER benutzt. Sie gibt den Wert FEHLER zurück, wenn der Benutzer keinen Wert in die Zelle AMT eingibt, und gibt den Inhalt von AMT (einen Wert) zurück, wenn der Benutzer einen Wert eingibt.

```
@WENN(@ZELLE("Typ";AMT)="w";AMT;@FEHLER)
```

Im folgenden Beispiel wird @ZELLZEIGER in einem Makro benutzt, mit dem in einer Liste mit Elementen auf eine leere Zelle geprüft wird. Wenn 1-2-3 im folgenden Beispiel auf eine leere Zelle stößt, löst es ein Akustiksignal aus und springt zu einem Unterprogramm.

```
{@WENN @ZELLZEIGER("Typ")="1"}{TON}SPRUNG Schritt2}
```

Ähnliche @Funktionen

@ gibt den Inhalt einer bestimmten Zelle indirekt wieder.

@ZEICHEN

@ZEICHEN(x) gibt das Zeichen aus dem Lotus Multibyte Zeichensatz (LMBCS) zurück, das der Zahl x entspricht.

Argumente

x ist eine Ganzzahl. Werte, die den LMBCS-Zeichencodes nicht entsprechen, ergeben FEHLER. Wenn x keine Ganzzahl ist, schneidet 1-2-3 sie auf eine Ganzzahl ab.

Hinweise

Können auf Ihrem Bildschirm die Zeichen nicht angezeigt werden, die x entsprechen, so zeigt 1-2-3 ein Zeichen an, das wenn möglich dem gewünschten Zeichen ähnlich ist. Entspricht kein Zeichen dem gewünschten Zeichen, so zeigt 1-2-3 ein ausgefülltes Rechteck an, das einem nicht anzeigbaren Zeichen entspricht. Prüfen Sie, ob Ihr Drucker die eingegebenen Zeichen ausdrucken kann.

Beispiele

@ZEICHEN(156) = £ (Britisches Pfund-Zeichen).

@ZEICHEN(D9) = A, wenn die Zelle D9 den Wert 65 enthält.

Ähnliche @Funktionen

@CODE gibt den LMBCS-Code zurück, der einem Zeichen entspricht.

@CHIVERTEILUNG

@CHIVERTEILUNG(x ;Freiheitsgrade;[Typ]) berechnet die Chi-Quadrat-Verteilung.

Argumente

x ist der Wert, mit dem die Chi-Quadrat-Verteilung ausgewertet werden soll. Der Wert, den Sie für x eingeben, hängt von dem Wert ab, den Sie für *Typ* eingeben.

Wenn *Typ* ist x

0 der kritische Wert oder obere Grenzwert für den Wert der Zufallsvariablen der kumulativen Chi-Quadrat-Verteilung. Hier handelt es sich um einen Wert größer als oder gleich 0; Vorgabe, wenn Sie kein Argument angeben

1 Eine Wahrscheinlichkeit (Signifikanzebene); entspricht einem Wert von 0 bis 1

Freiheitsgrade ist die Anzahl von Freiheitsgraden für die Stichprobe. *Freiheitsgrade* ist eine positive Ganzzahl. Wenn *Freiheitsgrade* keine Ganzzahl ist, schneidet 1-2-3 sie auf eine Ganzzahl ab.

Typ ist ein wahlweises Argument, mit dem angegeben wird, wie 1-2-3 @CHIVERTEILUNG berechnet.

| <i>Typ</i> | 1-2-3 berechnet |
|------------|--|
| 0 | Die Signifikanzebene, die x entspricht; Vorgabe, wenn Sie das Argument weglassen |
| 1 | Der kritische Wert, der der Signifikanzebene x entspricht |

Hinweise

@CHIVERTEILUNG berechnet die kumulative Chi-Quadrat-Verteilung näherungsweise innerhalb des Bereichs $\pm 3 \cdot 10^{-7}$. Wenn @CHIVERTEILUNG das Ergebnis nach 100 Neuberechnungsversuchen nicht innerhalb von 0,0000001 nähern kann, ist das Ergebnis FEHLER.

Die Chi-Quadrat-Verteilung ist eine kontinuierliche, ein-parametrische Verteilung, die als ein Sonderfall der Gamma-Verteilung abgeleitet wird.

Mit @CHIVERTEILUNG testen Sie die Gültigkeit einer Hypothese, indem Sie die beobachteten Werte mit den erwarteten Werten vergleichen.

Beispiele

@CHIVERTEILUNG(12,592;6) = 0,05

@CHIVERTEILUNG(0,05;6;1) = 12,59159

Ähnliche @Funktionen

@CHITEST berechnet die Wahrscheinlichkeit, die mit einem Chi-Quadrat-Test verknüpft ist. @FVERTEILUNG berechnet die F-Verteilung. @TVERTEILUNG berechnet die Studentsche t -Verteilung.

@WAHL

@WAHL(x;Argument-Liste) gibt einen mit x dargestellten Wert oder ein Label aus *Argument-Liste* zurück.

Argumente

x ist ein Wert. x entspricht dem Zählwert, der die Position eines Elements in *Argument-Liste* angibt.

Argument-Liste entspricht einer Gruppe von Werten bzw. den Adressen oder Namen von Zellen, die Werte und Labels enthalten, durch Argument-Trennzeichen getrennt. 1-2-3 numeriert jeden Eintrag in *Argument-Liste* und wählt dann den Eintrag, der dem Wert von x entspricht.

Beispiele

Ein Arbeitsblatt enthält eine Liste mit Labels im Bereich A1..A4 und eine Liste von deren Zählwerten (0, 1, 2, 3) im Bereich B1..B4. **@WAHL(B3;A1;A2;A3;A4)** gibt das Label in A3 zurück, d.h. das Element, dessen Zählwert 2 (2 ist der Wert in B3) in *Argument-Liste* entspricht.

Ähnliche @Funktionen

@HVERWEIS und **@VVERWEIS** ermitteln Einträge in horizontalen oder vertikalen Verweistabellen. **@INDEX** und **@XINDEX** geben den Inhalt einer Zelle bei dem Schnittpunkt zwischen einer angegebenen Spalte, einer angegebenen Zeile und einem angegebenen Arbeitsblatt zurück. **@VERGLEICH** gibt die Position der Zelle in einem Bereich zurück, deren Inhalt den von Ihnen angegebenen Daten entspricht. **@MAXVERWEIS** ergibt eine absolute Referenz auf die Zelle, die den größten Wert in einer Liste mit Bereichen enthält. **@MINVERWEIS** ergibt eine absolute Referenz auf die Zelle, die den kleinsten Wert in einer Liste mit Bereichen enthält.

@KLÄREN

@KLÄREN(*Zeichenfolge*) löscht nicht druckbare Zeichen aus *Zeichenfolge*.

Argumente

Zeichenfolge ist Text in " " (Anführungszeichen), die Adresse oder der Name einer Zelle, die ein Label enthält, bzw. eine Formel oder @Funktion, die Text ergibt.

Beispiele

Sie haben Daten aus einem Textverarbeitungsprogramm in 1-2-3 importiert. Zelle A45 enthält das Label

®Zweitens, wir müssen schnell reagieren.¬

@KLÄREN(A45) = Zweitens, wir müssen schnell reagieren.

Ähnliche @Funktionen

@ZEICHEN gibt das LMBCS-Zeichen zurück, das einer Codezahl entspricht. @KOMPR löscht führende, abschließende und fortlaufende Leerstellen aus Text.

@CODE

@CODE(*Zeichenfolge*) gibt den Code des Lotus Multibyte Zeichensatzes (LMBCS) zurück, der dem ersten Zeichen in *Zeichenfolge* entspricht.

Argumente

Zeichenfolge ist Text in " " (Anführungszeichen), die Adresse oder der Name einer Zelle, die ein Label enthält, bzw. eine Formel oder @Funktion, die eine Zeichenfolge ergibt.

Beispiele

@CODE("A") = 65.

@CODE(C5) = 77, wenn C5 das Label Mr. Jones enthält, da 77 der LMBCS-Code für M ist.

Ähnliche @Funktionen

@ZEICHEN gibt das LMBCS-Zeichen zurück, das einer Codezahl entspricht.

@SPALTEN

@SPALTEN(*Bereich*) zählt die Anzahl von Spalten in *Bereich*.

Argumente

Bereich ist eine Bereichadresse oder ein Bereichsname.

Hinweise

Benutzen Sie @SPALTEN mit {FÜR} in einem Makro, der denselben Schritt mit einer Reihe von Spalten ausführt, um festzulegen, wann der Makro gestoppt werden soll.

Beispiele

SPALTEN(D9..J25) = 7, da *Bereich* die Spalten D bis J (7 Spalten) umfaßt.

@SPALTEN(ERGEBNIS) = 2, wenn ERGEBNIS der Name des Bereichs B3..C45 ist.

Ähnliche @Funktionen

@REFKONVERT wandelt die 1-2-3 Spalten- oder Arbeitsblattbuchstaben A bis IV in Zahlen von 1 bis 256 um.

@ZEILEN zählt die Zeilen, und @BLÄTTER zählt die Arbeitsblätter in einem Bereich.

@KOMBIN

@KOMBIN($n;r$) berechnet den Binomialkoeffizienten für n und r . Der Binomialkoeffizient entspricht der Anzahl von Möglichkeiten, mit denen r aus n gewählt werden kann, ohne Berücksichtigung der Rangfolge.

Argumente

n entspricht der Anzahl von Werten und ist eine beliebige positive Ganzzahl oder 0.

r entspricht der Anzahl von Werten in jeder Kombination und ist eine beliebige positive Ganzzahl oder 0. r muß kleiner als oder gleich n sein.

Wenn es sich bei n und r nicht um Ganzzahlen handelt, schneidet 1-2-3 sie auf Ganzzahlen ab.

Hinweise

@KOMBIN berechnet den Binomialkoeffizienten näherungsweise innerhalb eines Bereichs von $\pm 3 \cdot 10^{-7}$.

Beispiele

Ein Gefäß enthält fünf Kugeln, jede in einer anderen Farbe. Sie nehmen drei beliebige Kugeln heraus. Die Anzahl von möglichen Farbkombinationen ist

@KOMBIN(5;3) = 10

Ähnliche @Funktionen

@BINOMIAL berechnet die binomiale Wahrscheinlichkeits-Massenfunktion oder die kumulative Binomialverteilung.

@KRITBINOMIAL berechnet den kleinsten Wert, für den die kumulative Binomialverteilung kleiner als oder gleich einem bestimmten Kriterium ist. @PERMUT berechnet die Anzahl von Permutationen für eine Werteliste.

@KOORD

@KOORD(*Arbeitsblatt*; *Spalte*; *Zeile*; *Absolut*) erstellt eine Zelladresse anhand von Werten, die *Arbeitsblatt*, *Spalte* und *Zeile* entsprechen.

Argumente

Arbeitsblatt und *Spalte* entsprechen beliebigen Ganzzahlen von 1 bis 256. *Arbeitsblatt* und *Spalte* entsprechen den Arbeitsblatt- und Spaltenbuchstaben (1 für Arbeitsblatt oder Spalte A, 2 für Arbeitsblatt oder Spalte B usw.).

Zeile ist eine beliebige Ganzzahl von 1 bis 8192. *Zeile* entspricht der Zeilennummer.

Absolut ist eine beliebige Ganzzahl von 1 bis 8.

Wenn *Arbeitsblatt*, *Spalte*, *Zeile* und *Absolut* keine Ganzzahlen sind, schneidet 1-2-3 diese auf Ganzzahlen ab.

Hinweise

@KOORD erstellt je nach Wert von *Absolut* eine relative, absolute oder gemischte Zelladresse. In der folgenden Tabelle werden die möglichen Werte von *Absolut* und deren Auswirkung auf die Zelladresse A1 in Arbeitsblatt A dargestellt.

| <i>Absolut</i> | Wert von @KOORD(1;1;1; <i>Absolut</i>) |
|----------------|---|
| 1 | \$A:\$A\$1 |
| 2 | \$A:A\$1 |
| 3 | \$A:\$A1 |
| 4 | \$A:A1 |
| 5 | A:\$A\$1 |
| 6 | A:A\$1 |
| 7 | A:\$A1 |
| 8 | A:A1 |

Benutzen Sie @KOORD mit @INDEX, @VVERWEIS oder @HVERWEIS, um Zelladressen aus Wertetabellen in der aktuellen Arbeitsmappe zu erstellen. Benutzen Sie @KOORD mit @@, um den Wert in der Zelladresse zurückzugeben, die mit @KOORD erstellt wurde.

Beispiele

@KOORD(3;7;25;8) gibt die relative Zelladresse C:G25 zurück.

@@(@KOORD(C1;D1;E4;8)) gibt den Wert in Zelle A:A4 zurück (C1 enthält 1, D1 enthält 1 und E4 enthält 4).

Ähnliche @Funktionen

@REFKONVERT wandelt die 1-2-3 Spalten- oder Arbeitsblattbuchstaben A bis IV in Zahlen von 1 bis 256 um.

@KORREL

@KORREL(*Bereich1*;*Bereich2*) berechnet den Korrelationskoeffizienten von Werten in *Bereich1* und *Bereich2*.

Argumente

Bereich1 und *Bereich2* sind Bereichsnamen oder Bereichsadressen. *Bereich1* und *Bereich2* müssen dieselbe Größe haben. Wenn *Bereich1* und *Bereich 2* nicht dieselbe Größe haben, ergibt @KORREL den Wert FEHLER.

1-2-3 ordnet die Zellen in den beiden Bereichen je nach Rangfolge paarweise an. Die Bereiche werden von oben nach unten, von links nach rechts, vom ersten zum letzten Blatt geordnet.

Hinweise

Korrelation und Kovarianz messen beide die Beziehung zwischen zwei Datengruppen. Die Korrelationsstatistik ist jedoch von der Maßeinheit unabhängig, während die Kovarianzstatistik von der Maßeinheit abhängig ist.

Beispiele

@KORREL

Ähnliche @Funktionen

@KOVAR berechnet die Kovarianz der Werte in zwei Bereichen.

@COS

@COS(z) berechnet den Kosinus des Winkels z. Der Kosinus ist gleich der Ankathete dividiert durch die Hypotenuse. Das Ergebnis von @COS ist ein Wert von -1 bis 1.

Argumente

z ist ein Winkel gemessen in Radian. z ist ein beliebiger Wert von von -2π bis 2π .

Beispiele

@COS(@GRADINRAD(30)) = 0,866, der Kosinus eines Winkels von 30 Grad.

Ähnliche @Funktionen

@ARCCOS berechnet den Arkuskosinus eines Wertes. @COSH berechnet den hyperbolischen Kosinus eines Wertes.

@COSH

@COSH(z) berechnet den hyperbolischen Kosinus des Winkels z. Das Ergebnis von @COSH ist ein Wert größer als oder gleich 1.

Argumente

z ist ein beliebiger Wert von etwa 11355,1371 bis 11355,1371.

Beispiele

@COSH(@GRADINRAD(30)) = 1,140238, der hyperbolische Kosinus eines Winkels von 30 Grad.

Ähnliche @Funktionen

@ARCCOS berechnet den Arkuskosinus (inversen Kosinus) eines Wertes. @COS berechnet den Kosinus eines Winkels.

@COT

@COT(z) berechnet den Kotangens des Winkels z. Der Kotangens ist gleich der Ankathete dividiert durch die gegenüberliegende Seite.

Argumente

z ist ein Winkel gemessen in Radian. z kann ein beliebiger Wert von -2^63 bis 2^63 sein .

Beispiele

@COT(@GRADINRAD(30)) = 1,73205, der Kotangens eines Winkels mit 30 Grad.

Ähnliche @Funktionen

@ARCCOT berechnet den Arkuskotangens eines Wertes. @COTH berechnet den hyperbolischen Kotangens eines Winkels. @TAN berechnet den Tangens eines Winkels.

@COTH

@COTH(z) berechnet den hyperbolischen Kotangens des Winkels z.

Argumente

z ist ein beliebiger Wert von zwischen -709,7827 und 709,7827.

Beispiele

@COTH(@GRADINRAD(30)) = 2,081283, der hyperbolische Kotangens eines Winkels von 30 Grad.

Ähnliche @Funktionen

@ARCOTH berechnet den hyperbolischen Arkuskotangens eines Wertes. @COT berechnet den Kotangens eines Winkels. @TANH berechnet den hyperbolischen Tangens eines Winkels.

@ANZAHL, @PURANZAHL

@ANZAHL(*Argument-Liste*) zählt die nichtleeren Zellen in einer *Liste* mit Bereichen.

@PURANZAHL(*Argument-Liste*) zählt die Zellen in einer *Liste* mit Bereichen, wobei die Zellen, die Label enthalten, von der Zählung ausgeschlossen werden.

Argumente

Argument-Liste ist eine beliebige Kombination aus Bereichsadressen oder Bereichsnamen, durch Argument-Trennzeichen getrennt.

Siehe auch Argumente für statistische @Funktionen.

Hinweise

@ANZAHL zählt jede Zelle in *Argument-Liste*, die einen beliebigen Eintrag enthält, einschließlich einem Label, einem Justierungszeichen oder den Werten FEHLER und NV.

@ANZAHL und @PURANZAHL werden benutzt, um einen Makro zu stoppen (oder umzuleiten), der eine Funktion mit einer Reihe von Bereichen ausführt, wenn der Zellzeiger einen Bereich erreicht, der keine Einträge enthält.

Beispiele

@ANZAHL(A2..A3;A5) = 1, wenn A2..A3 leer ist und unabhängig davon, ob A5 leer ist oder nicht, da A5 eine einzellige Adresse ist.

{@WENN @PURANZAHL(SEPTEMBER)=0}{SPRUNG BIS_DATO} springt zu einem Makro namens BIS_DATO, wenn der Bereich SEPTEMBER leer ist oder ein Label, ein Justierungszeichen oder eine Textformel enthält.

Ähnliche @Funktionen

@DANZAHL und @DPURANZAHL zählt die nichtleeren Zellen in einem Feld einer Datenbanktabelle, die den angegebenen Kriterien entsprechen.

@KOVAR

@KOVAR(Bereich1;Bereich2;[Typ]) berechnet die Kovarianz einer Grundgesamtheit oder Stichprobe der Werte in *Bereich1* und *Bereich2*.

Argumente

Bereich1 und *Bereich2* sind die Namen oder Adressen von Bereichen. *Bereich1* und *Bereich2* müssen dieselbe Größe haben. Wenn *Bereich1* oder *Bereich 2* nicht die gleiche Größe haben, ergibt @KOVAR den Wert FEHLER.

1-2-3 ordnet die Zellen in den beiden Bereichen paarweise nach derselben Rangfolge im Bereich zu. Bereiche werden von oben nach unten, links nach rechts und vom ersten zum letzten Blatt geordnet.

Typ ist ein wahlweises Argument, mit dem angegeben wird, ob die Kovarianz einer Grundgesamtheit oder einer Stichprobe berechnet werden soll.

| | |
|------------|---|
| Typ | 1-2-3 berechnet |
| 0 | Kovarianz einer Grundgesamtheit; Vorgabe, wenn Sie das Argument weglassen |
| 1 | Kovarianz einer Stichprobe |

Hinweise

Die Kovarianz ist der Mittelwert der Produkte der Abweichung entsprechender Werte in Listen.

Korrelation und Kovarianz messen beide die Beziehung zwischen zwei Datengruppen. Die Korrelationsstatistik ist jedoch von der Maßeinheit unabhängig, während die Kovarianzstatistik von der Maßeinheit abhängig ist.

Beispiele

@KOVAR

Ähnliche @Funktionen

@VAR und @PURVAR berechnen die Varianz einer Grundgesamtheit, während @VARP und @PURVARP die Varianz einer Stichprobe der Werte in einer Liste berechnen. @KORREL berechnet den Korrelationskoeffizienten entsprechender Werte in zwei Bereichen.

@COSEC

@COSEC(z) berechnet den Kosekans des Winkels z. Der Kosekans ist der Reziprokwert des Sinus. Das Ergebnis von @COSEC ist ein Wert größer als oder gleich 1 bzw. kleiner als oder gleich -1.

Argumente

z ist ein Winkel gemessen in Radian. z ist ein beliebiger Wert von $-2^{32} \cdot \pi$ bis $2^{32} \cdot \pi$, mit Ausnahme von 0 und Vielfachen von π .

Beispiele

@COSEC(@GRADINRAD(30)) = 2, der Kosekans eines Winkels mit 30 Grad.

Ähnliche @Funktionen

@ARCCOSEC berechnet den Arkuskosekans eines Wertes. @COSECH berechnet den hyperbolischen Kosekans eines Winkels. @SIN berechnet den Sinus eines Winkels.

@COSECH

@COSECH(z) berechnet den hyperbolischen Kosekans des Winkels z. Der hyperbolische Kosekans ist der Reziprokwert des hyperbolischen Sinus.

Argumente

z ist ein beliebiger Wert von etwa 11355,1371 bis 11355,1371, mit Ausnahme von 0.

Beispiele

@COSECH(@GRADINRAD(30)) = 1,825306, der hyperbolische Kosekans eines Winkels mit 30 Grad.

Ähnliche @Funktionen

@ARCOSECH berechnet den hyperbolischen Arkuskosekans eines Wertes. @COSEC berechnet den Kosekans eines Winkels. @SINH berechnet den hyperbolischen Sinus eines Winkels.

@LAUF

@LAUF(*Zinssatz;Zukwert;Aktwert*) berechnet die Anzahl von Perioden, bis eine Investition (*Aktwert*) auf einen zukünftigen Wert anwächst, wobei ein fester *Zinssatz* pro Zahlungsperiode fortgeschrieben wird.

Argumente

Zinssatz ist ein beliebiger Wert größer als -1, mit Ausnahme von 0.

Zukwert und *Aktwert* sind beliebige Werte. *Zukwert* und *Aktwert* müssen entweder positiv oder negativ sein.

Beispiele

Sie haben gerade 10.000 DM auf ein Konto einbezahlt, für das Sie einen jährlichen Zinssatz von 10% (0,10) erhalten, der monatlich bezahlt wird. Nun möchten Sie feststellen, wie viele Jahre es dauert, bis sich Ihre Investition verdoppelt hat.

@LAUF(0,10/12;20000;10000)/12 = 6,960312.

Anders ausgedrückt, es dauert etwa sieben Jahre, bis sich Ihre ursprüngliche Investition von 10.000 DM verdoppelt hat.

Mit **@LAUF** berechnen Sie die Gesamtanzahl der Zahlungsperioden. Deshalb müssen Sie unter Umständen die Anzahl von Zahlungsperioden angeben, in denen der *Zinssatz* berechnet wird, damit die Laufzeit und der Zinssatz in derselben Zeiteinheit angegeben werden. In dem obigen Beispiel beträgt der jährliche Zinssatz 10%, wird monatlich bezahlt und wird als 0,10/12 eingegeben (*Zinssatz* wird durch die Anzahl von Zahlungsperioden dividiert).

Ähnliche @Funktionen

@ANN und **@NPER** errechnen die Anzahl von Perioden, die eine Investition mit gleichen periodischen Zahlungen benötigt, um einen bestimmten Wert zu erreichen.

Beispiel: @@

In dieser einfachen Grafik für Umsatzprovisionen enthält die Zelle A10 die Formel

`@WENN(C3="S";"C7";@WENN(C3="D";"C8";@FEHLER))`

die zu einer von zwei Zelladressen (C7 oder C8) führt, je nachdem, welcher Produktcode (S oder D) in C3 eingegeben wurde. Wenn Sie @@(A10) in C4 eingeben, wird der Inhalt der Zelle zurückgegeben, deren Adresse von der Formel in A10 berechnet wird.

Wenn Sie in C3 eine andere Angabe als einen der beiden Produktcodes machen, ergeben sowohl die @WENN als auch die @@ Funktion FEHLER.

A ----- A ----- B ----- C -----

1 UMSATZPROVISIONS-GRAFIK

2

3 Produktcode eingeben: S

4 Provisionssatz: 5%

5

| 6 Produkt | Code | Satz |
|-----------|------|------|
|-----------|------|------|

| | | |
|-------------|---|----|
| 7 Schrauben | S | 5% |
|-------------|---|----|

| | | |
|--------------|---|----|
| 8 Dichtungen | D | 3% |
|--------------|---|----|

9

10 C7

Beispiel: @MITTELABW

In dieser Tabelle werden die Hausverkäufe im Monat April aufgeführt. Das Alter (in Jahren) der Häuser wird in einem Bereich namens ALTER(C2..C8) aufgeführt. Nun möchten Sie die mittlere Abweichung des Alters der Häuser in der Tabelle ermitteln.

@MITTELABW(ALTER) = 15,18

| A | A | B | C | D |
|---|-----------------|--------|-------|------------|
| 1 | ADRESSE | HEKTAR | ALTER | PREIS |
| 2 | Kaiserstraße 12 | 0,25 | 48 | DM 290.000 |
| 3 | Liststraße | 0,40 | 22 | DM 105.000 |
| 4 | Königstraße 103 | 0,50 | 21 | DM 135.000 |
| 5 | Bergstraße 27 | 1,00 | 70 | DM 128.000 |
| 6 | Arminstraße 46 | 0,50 | 52 | DM 174.000 |
| 7 | Fasanenstraße 9 | 0,25 | 42 | DM 195.000 |
| 8 | Strandstraße 80 | 0,25 | 23 | DM 118.000 |

Beispiel: @KORREL

Sie möchten feststellen, ob zwischen Größe und Gewicht von zehn zufällig gewählten Testpersonen eine Beziehung besteht. Sie schreiben Größe und Gewicht der Personen in einem Arbeitsblatt auf.

@KORREL(A2..A11;B2..B11) = 0,384947

| | A ----- | A ----- | B -- |
|----|-------------|--------------|------|
| 1 | GRÖSSE (cm) | GEWICHT (kg) | |
| 2 | 190,50 | 72,73 | |
| 3 | 187,96 | 86,36 | |
| 4 | 175,26 | 68,18 | |
| 5 | 175,26 | 76,37 | |
| 6 | 180,34 | 77,27 | |
| 7 | 180,34 | 72,73 | |
| 8 | 187,96 | 75,00 | |
| 9 | 172,72 | 68,18 | |
| 10 | 177,80 | 70,46 | |
| 11 | 179,07 | 86,36 | |

Beispiel: @KOVAR

Sie möchten feststellen, ob zwischen der Größe und dem Gewicht von zehn zufällig gewählten Testpersonen eine Beziehung besteht. Sie schreiben Größe und Gewicht der Personen in einem Arbeitsblatt auf.

@KOVAR(A2..A11;B2..B11) = 13,8872

A ----- A ----- B --

| 1 | GRÖSSE (cm) | GEWICHT (kg) |
|----|-------------|--------------|
| 2 | 190,50 | 72,73 |
| 3 | 187,96 | 86,36 |
| 4 | 175,26 | 68,18 |
| 5 | 175,26 | 76,37 |
| 6 | 180,34 | 77,27 |
| 7 | 180,34 | 72,73 |
| 8 | 187,96 | 75,00 |
| 9 | 172,72 | 68,18 |
| 10 | 177,80 | 70,46 |
| 11 | 179,07 | 86,36 |

@DATUM

@DATUM(*Jahr;Monat;Tag*) berechnet die Datumseriennummer für das angegebene *Jahr*, den angegebenen *Monat* und den angegebenen *Tag*.

Argumente

Jahr ist eine Ganzzahl von 0 (das Jahr 1900) bis 199 (das Jahr 2099).

Monat ist eine Ganzzahl von 1 bis 12.

Tag ist eine Ganzzahl von 1 bis 31. Der für *Tag* benutzte Wert muß ein gültiger Tag für den Monat sein. So können Sie z.B. 31 nicht als *Tag* benutzen, wenn Sie 4 (April) als *Monat* benutzen.

Hinweise

Obwohl es keinen 29. Februar 1900 gab (1900 war kein Schaltjahr), weist 1-2-3 diesem Tag eine Datumseriennummer zu. Dadurch wird keine der Datumsberechnungen ungültig, es sei denn, Sie benutzen Datumsangaben vom 1. Januar 1900 bis zum 1. März 1900. Wenn Sie Datumsangaben innerhalb dieses Zeitraums verwenden, subtrahieren Sie von jedem Ergebnis innerhalb dieses Zeitraums 1.

Wenn die Ergebnisse einer **@DATUM** Berechnung als richtige Datumsangabe angezeigt werden sollen, formatieren Sie die Zelle, in der die **@DATUM** Funktion steht, mit einem der Datumsformat Datumsformate.

Beispiele

@DATUM(92;2;21) ergibt 33655 oder 21-Feb-92 in einer als Tag-Monat-Jahr formatierten Zelle.

@DATUM(91;2;29) ergibt FEHLER, da 1991 kein Schaltjahr war.

Ähnliche @Funktionen

@DATUMWERT berechnet die Datumseriennummer für ein als Label eingegebenes Datum. **@ZEIT** berechnet die Zeitseriennummer für eine bestimmte Zeit. **@JETZI** berechnet die Datum- und Zeitseriennummer für das laufende Datum und die laufende Uhrzeit.

@DATUMDIFF

DATUMDIFF(*Anfangsdatum*; *Enddatum*; *Format*) berechnet die Anzahl von Jahren, Monaten oder Tagen zwischen zwei Datumseriennummern.

Argumente

Anfangsdatum und *Enddatum* sind Datumseriennummern.

Format ist ein Code aus der folgenden Tabelle, als Text eingegeben, mit dem das Format angegeben wird, in dem das Ergebnis von @DATUMDIFF ausgegeben werden soll.

| Format | Anzeige |
|---------------|---|
| j | Jahre |
| m | Monate |
| t | Tage |
| mt | Tage; Monate und Jahre werden ignoriert |
| jm | Monate; Jahre werden ignoriert |
| jt | Tage; Jahre werden ignoriert |

Beispiele

In den folgenden Beispielen werden die Datumsangaben 15. Februar 1990 und 15. September 1993 benutzt.

@DATUMDIFF(@DATUM(90;2;15);@DATUM(93;9;15);"m") ergibt 43, die Anzahl von Monaten zwischen dem 15. Februar 1990 und dem 15. September 1993.

@DATUMDIFF(@DATUM(90;2;15);@DATUM(93;9;15);"mt") ergibt 0, da der Tag des Monats sowohl bei *Anfangs-* als auch bei *Enddatum* der 15. ist.

@DATUMDIFF(@DATUM(90;2;15);@DATUM(93;9;15);"jm") ergibt 7, die Anzahl von Monaten zwischen Februar und September.

Ähnliche @Funktionen

@D360 und @DAYS360 berechnen die Anzahl von Tagen zwischen zwei Datumsangaben auf der Grundlage eines 360-Tage-Jahres (12 Monate mit jeweils 30 Tagen). @TAGE berechnet die Anzahl von Tagen zwischen zwei Datumsangaben auf der Grundlage einer von Ihnen angegebenen Anzahl von Tagen im Jahr.

@DATUMINFO

@DATUMINFO(*Datum*;*Attribut*) gibt Informationen über eine Datumseriennummer zurück.

Argumente

Datum ist eine Datumseriennummer.

Attribut ist eine der Ganzzahlen, die in der folgenden Tabelle aufgeführt werden:

| Attribut | Ergebnis |
|-----------------|--|
| 1 | Wochentag als Label, Kurzformat (Mon) |
| 2 | Wochentag als Label, langes Format (Montag) |
| 3 | Wochentag als Ganzzahl von 0 (Montag) bis 6 (Sonntag) |
| 4 | Woche innerhalb des Jahres als Ganzzahl von 1 bis 53 |
| 5 | Monat des Jahres als Label, Kurzformat (Jan) |
| 6 | Monat des Jahres als Label, langes Format (Januar) |
| 7 | Anzahl von Tagen im Monat, wie vom <i>Datum</i> angegeben |
| 8 | Restliche Anzahl von Tagen im Monat, wie mit <i>Datum</i> angegeben |
| 9 | Datumseriennummer für den letzten Tag des Monats, wie mit <i>Datum</i> angegeben |
| 10 | Das Quartal, in dem die <i>Datumsangabe</i> liegt, als Ganzzahl von 1 (Q1) bis 4 (Q4) |
| 11 | 1, wenn das mit <i>Datum</i> angegebene Jahr ein Schaltjahr ist; 0, wenn das Jahr kein Schaltjahr ist. |
| 12 | Tag des Jahres, wie mit <i>Datum</i> angegeben, als Zahl von 1 bis 366 |
| 13 | Restliche Tage im Jahr, wie mit <i>Datum</i> angegeben, als Zahl |

Beispiele

@DATUMINFO(23063;7) = 28, die Anzahl von Tagen im Februar 1963.

@DATUMINFO(@DATUM(92;10;5);10) = 4, da der Oktober im vierten Quartal liegt.

@DATUMWERT

@DATUMWERT(*Zeichenfolge*) berechnet die Datumseriennummer für das in *Zeichenfolge* angegebene Datum.

Argumente

Zeichenfolge ist eine Zeichenfolge in " " (Anführungszeichen), eine Formel, die eine Zeichenfolge ergibt, oder die Adresse bzw. der Name einer Zelle, die ein Label oder eine Formel enthält, die ein Label ergibt. *Zeichenfolge* muß eines der folgenden Formate haben:

- Ein Datumsformat, das in der Liste "Häufig verwendet" aufgeführt wird
- Eines der fünf Datumsformate, die in Zellen mit dem Standardformat akzeptiert werden: 31-Dez-96, Dez-96, 31-Dez, und die in den Ländereinstellungen des Betriebssystems festgelegten kurzen und langen Datumsformate.

Hinweise

Wenn die Ergebnisse einer @DATUMWERT Berechnung in Form eines tatsächlichen Datums angezeigt werden sollen, formatieren Sie die Zelle, in der die @DATUMWERT Funktion steht, mit einem der Datumsformate.

@DATUMWERT wird besonders mit Daten benutzt, die aus einem anderen Programm importiert wurden, wie z.B. einem Textverarbeitungsprogramm.

Beispiele

@DATUMWERT("21-Feb-91") ergibt die Datumseriennummer 33290.

@DATUMWERT(GEBURTSTAG) ergibt die Datumseriennummer 20723, wenn die Zelle GEBURTSTAG das Label 25-Sep-56 enthält.

Ähnliche @Funktionen

@DATUMFOLGE wandelt eine Datumseriennummer in das entsprechende Datum um und zeigt sie als Label an.

@DATUM berechnet die Datumseriennummer für ein bestimmtes Datum.

@TAG

@TAG(*Datumseriennummer*) berechnet den Tag des Monats, d.h. einen Wert von 1 bis 31, aus einer *Datumseriennummer*.

Argumente

Datumseriennummer ist ein Wert von 1 (1. Januar 1900) bis 73050 (31. Dezember 2099).

Hinweise

Mit einer der anderen Datum-@Funktionen können Sie den Wert für Datumseriennummer angeben.

@TAG kann das *Tag*-Argument für andere Datum-@Funktionen liefern, die auf vorher berechneten Datumsangaben beruhen.

Beispiele

@TAG(@JETZT) = der laufende Tag des Monats.

@TAG(D9) = 12, wenn Zelle D9 die Datumseriennummer 33250 (das Datum 12-Jan-91) enthält.

Ähnliche @Funktionen

@MONAT berechnet den Monat, @JAHR berechnet das Jahr, und @WOCHENTAG berechnet den Wochentag aus einer Datumseriennummer.

@TAGE

@TAGE(*Anfangsdatum*; *Enddatum*; [*Basis*]) berechnet die Anzahl von Tagen zwischen zwei Datumsangaben mit einer angegebenen *Basis* für die Zählung der Tage.

Argumente

Anfangsdatum und *Enddatum* sind Datumseriennummern. Wenn *Anfangsdatum* vor *Enddatum* liegt, ist das Ergebnis von @TAGE positiv. Wenn *Anfangsdatum* später als *Enddatum* ist, ist das Ergebnis von @TAGE negativ. Wenn *Anfangsdatum* und *Enddatum* identisch sind, ist das Ergebnis von @TAGE gleich 0.

Basis ist ein wahlweises Argument, mit dem angegeben wird, auf welcher Basis die Tage gezählt werden sollen.

Basis ist ein Wert aus der folgenden Tabelle:

| <u>Basis</u> | <u>Basis für das Zählen der Tage</u> |
|--------------|--|
| 0 | 30/360; Vorgabe, wenn Sie das Argument weglassen |
| 1 | Anzah/Anzahl |
| 2 | Anzahl/360 |
| 3 | Anzahl/365 |
| 4 | Europäisch 30/360 |

Beispiele

@TAGE(@DATUM(93;4;16);@DATUM(93;9;25)) = 159, die Anzahl von Tagen zwischen dem 16. April 1993 und dem 25. September 1993 auf der Grundlage eines 360-Tage-Jahres mit 12 Monaten zu jeweils 30 Tagen.

@TAGE(@DATUM(93;4;16);@DATUM(93;9;25);1) = 162, die Anzahl von Tagen zwischen dem 16. April 1993 und dem 25. September 1993 auf der Grundlage der tatsächlichen Anzahl von Tagen in den Monaten April bis September.

Ähnliche @Funktionen

@DATUMDIFF berechnet die Anzahl von Jahren, Monaten oder Tagen zwischen zwei Datumsangaben. @D360 und @DAYS360 berechnen die Anzahl von Tagen zwischen zwei Datumsangaben auf der Grundlage eines 360-Tage-Jahres. @NETTOARBEITSTAGE berechnet die Anzahl von Arbeitstagen zwischen zwei Datumsangaben, Wochenenden und Feiertage ausgenommen.

@D360, @DAYS360

@D360(*Anfangsdatum*; *Enddatum*) berechnet die Anzahl von Tagen zwischen zwei Datumsangaben auf der Grundlage eines 360-Tage-Jahres (12 Monate mit jeweils 30 Tagen).

@DAYS360(*Anfangsdatum*; *Enddatum*) berechnet die Anzahl von Tagen zwischen zwei Datumsangaben auf der Grundlage eines 360-Tage-Jahres entsprechend dem Standard der U.S. Securities Industry.

Argumente

Anfangsdatum und *Enddatum* sind Datumseriennummern.

Hinweise

Die Formel, mit der @DAYS360 berechnet wird, entspricht den Änderungen des Jahres 1990 der Securities Industry Association's Ausgabe der *Standard Security Calculation Methods* aus dem Jahr 1986.

@DAYS360 und @D360 ergeben im allgemeinen unterschiedliche Antworten für dieselben Daten, wenn das *Anfangsdatum* oder das *Enddatum* der letzte Tag des Monats ist.

Beispiele

@DAYS360(@DATUM(89;4;16);@DATUM(89;9;25)) = 159

@D360(33290;33524) = 232, die Anzahl von Tagen zwischen dem 21. Februar 1991 und dem 13. Oktober 1991 auf der Grundlage eines 360-Tage-Jahres.

Ähnliche @Funktionen

@DATUMDIFF berechnet die Anzahl von Jahren, Monaten oder Tagen zwischen zwei Datumsangaben. @TAGE berechnet die Anzahl von Tagen zwischen zwei Datumsangaben, wobei eine bestimmte Zählbasis benutzt wird.

@NETTOARBEITSTAGE berechnet die Anzahl von Arbeitstagen zwischen zwei Datumsangaben, Wochenenden und Feiertage ausgenommen. @ARBEITSTAG berechnet das Datum, das eine bestimmte Anzahl von Tagen vor oder nach einem bestimmten Datum liegt, Wochenenden und Feiertage ausgenommen. @NÄCHSTMONAT berechnet das Datum, das eine bestimmte Anzahl von Monaten vor oder nach einem bestimmten Datum liegt.

@AFAEDEG

@AFAEDEG(*Kosten*; *Restwert*; *Lebensdauer*; *Zeitraum*) berechnet die Abschreibung eines Anlagegutes mit einem Anfangswert *Kosten*, einer bestimmten *Lebensdauer* und einem *Restwert* für einen bestimmten *Zeitraum* mit der festen degressiven Abschreibung.

Argumente

Kosten entspricht dem Betrag, der für das Anlagegut bezahlt wurde. *Kosten* ist ein beliebiger positiver Wert oder 0. Wenn *Kosten* gleich 0 ist, ist das Ergebnis von **@AFAEDEG** gleich 0.

Restwert entspricht dem geschätzten Wert des Anlagegutes nach Ablauf der *Lebensdauer*. *Restwert* ist ein beliebiger positiver Wert oder 0. Wenn *Restwert* größer ist als *Kosten*, ist das Ergebnis von **@AFAEDEG** negativ.

Lebensdauer entspricht der Anzahl von Zeiträumen bis zur Abschreibung eines Anlagegutes auf den *Restwert*.

Lebensdauer ist ein beliebiger Wert größer als oder gleich 1 und kleiner als oder gleich *Lebensdauer*.

Zeitraum entspricht dem Zeitraum, für den der Abschreibungsbetrag ermittelt werden soll. *Zeitraum* ist ein beliebiger Wert größer als oder gleich 1.

Lebensdauer und *Zeitraum* müssen in denselben Einheiten ausgedrückt werden, im allgemeinen in Jahren.

Hinweise

Mit der festen degressiven Abschreibung wird die Abschreibungsrate im Vergleich zur geometrisch degressiven Abschreibung verlangsamt, so daß in späteren Zeiträumen ein höherer Abschreibungsaufwand entsteht. Die Abschreibung ist abgeschlossen, wenn der Buchwert des Anlagegutes - d.h. die Gesamtkosten des Anlagegutes minus der gesamten Abschreibung über alle vorherigen Zeiträume - den Restwert erreicht.

Beispiele

Sie haben gerade eine Büromaschine für 10.000 DM gekauft. Die Lebensdauer dieser Maschine beläuft sich auf acht Jahre, und der Restwert nach acht Jahren beträgt 1.200 DM. Nun möchten Sie die Abschreibung für das fünfte Jahr berechnen.

@AFAEDEG(10000;1200;8;5) = 806,51 DM

Ähnliche @Funktionen

@AFADEGV berechnet die Abschreibung mit der variablen degressiven Abschreibungsmethode. **@AFADEG** benutzt die geometrisch-degressive Abschreibungsmethode, **@AFALIN** benutzt die lineare Abschreibungsmethode und **@AFADIG** die digitale Abschreibungsmethode.

@AFADEG

@AFADEG(*Kosten*; *Restwert*; *Lebensdauer*; *Zeitraum*) berechnet die Abschreibung eines Anlagegutes mit einem Anfangswert *Kosten*, einer bestimmten *Lebensdauer* und einem *Restwert* für einen bestimmten *Zeitraum* mit der *geometrisch-degressiven Abschreibung*.

Argumente

Kosten entspricht dem Betrag, der für das Anlagegut bezahlt wurde. *Kosten* ist ein Wert größer oder gleich *Restwert*.

Restwert entspricht dem geschätzten Wert des Anlagegutes nach Ablauf der *Lebensdauer*. *Restwert* ist ein beliebiger Wert.

Lebensdauer entspricht der Anzahl von Zeiträumen bis zur Abschreibung eines Anlagegutes auf den *Restwert*.

Lebensdauer ist ein beliebiger Wert größer als 2.

Zeitraum entspricht dem Zeitraum, für den der Abschreibungsbetrag ermittelt werden soll. *Zeitraum* ist ein beliebiger Wert größer als oder gleich 1.

Sie müssen *Lebensdauer* und *Zeitraum* in denselben Einheiten ausdrücken, im allgemeinen in Jahren.

Hinweise

Bei der geometrisch-degressiven Abschreibung wird in früheren Zeiträumen mehr abgeschrieben als in späteren. Die Abschreibung ist beendet, wenn der Buchwert des Anlagegutes - die Gesamtkosten des Anlagegutes minus der gesamten Abschreibung über alle vorausgegangenen Zeiträume - den Restwert erreicht.

Wenn der Restwert eines Anlagegutes relativ gering ist, kann das Anlagegut möglicherweise mit **@AFADEG** nicht voll bis zum Ende der geschätzten Lebensdauer abgeschrieben werden. In diesem Fall könnten Sie **@AFADEGV** benutzen, mit der das Anlagegut immer voll innerhalb der geschätzten Lebensdauer abgeschrieben wird.

Beispiele

Sie haben gerade eine Büromaschine für 10.000 DM gekauft. Die Lebensdauer dieser Maschine beläuft sich auf acht Jahre, und der Restwert nach acht Jahren ist 1.200 DM. Nun möchten Sie den Abschreibungsaufwand für das fünfte Jahr mit der geometrisch-degressiven Abschreibung ermitteln.

@AFADEG(10000;1200;8;5) = 791,02 DM.

Ähnliche @Funktionen

@AFADEGV berechnet die Abschreibung mit der variablen degressiven Abschreibungsmethode. **@AFADEG** benutzt die feste degressive Abschreibungsmethode, **@AFALIN** benutzt die lineare Abschreibungsmethode und **@AFADIG** die digitale Abschreibungsmethode.

@DEZIMAL

@DEZIMAL(*Hex-Zeichenfolge*) wandelt einen Hexadezimalwert in den entsprechenden Dezimalwert mit Vorzeichen versehen um.

Argumente

Hex-Zeichenfolge ist ein Wert von 00000000 bis FFFFFFFF als Text eingegeben. *Hex-Zeichenfolge* kann maximal acht Zeichen umfassen und darf nur Zahlen von 0 bis 9 enthalten sowie Buchstaben von A bis F. Die Buchstaben können als Groß- oder Kleinbuchstaben angegeben werden.

Hinweise

Hexadezimalwerte von 00000000 bis 7FFFFFFF entsprechen 0 und positiven Dezimalwerten.

Hexadezimalwerte von 80000000 bis FFFFFFFF entsprechen negativen Dezimalwerten.

Beispiele

@DEZIMAL("1A") = 26

@DEZIMAL("FFFFFFFE") = -2

Ähnliche @Funktionen

@HEX wandelt Dezimalzahlen in Hexadezimalzahlen um.

@GRADINRAD

@GRADINRAD(*Grad*) wandelt *Grad* in Radian um.

Argumente

Grad ist ein Wert.

Beispiele

@GRADINRAD(30) = 0,523599 Radian

@COS(@GRADINRAD(45)) = 0,707107, der Kosinus eines Winkels von 45 Grad.

Ähnliche @Funktionen

@RADINGRAD wandelt Radian in Grad um.

@ABWQUADRAT

@ABWQUADRAT(*Argument-Liste*) berechnet die Summe der quadratischen Abweichung der Werte in *Argument-Liste* vom Mittel.

Argumente

Argument-Liste kann jede der folgenden Angaben in einer beliebigen Kombination enthalten: Zahlen, numerische Formeln und Adressen oder Namen von Bereichen, die Zahlen oder numerische Formeln enthalten. Trennen Sie die Elemente in der *Argument-Liste* mit Argument-Trennzeichen.

Siehe auch Argumente für statistische @Funktionen.

Beispiele

@ABWQUADRAT(2;3;9;8;15;2;1) = 159,4286

Ähnliche @Funktionen

@STDABW und @PURSTDABW berechnen die Standardabweichung der Werte in einer Argument-Liste.

@FEHLERF

@FEHLERF(*Untergrenze*;[*Obergrenze*]) berechnet die Fehlerfunktion zwischen *Untergrenze* und *Obergrenze*.

Argumente

Untergrenze ist die Untergrenze zur Integration von @FEHLERF und kann einen beliebigen Wert darstellen.

Obergrenze ist ein wahlweises Argument, mit dem die Obergrenze zur Integration von @FEHLERF angegeben wird. Wenn Sie das Argument *Obergrenze* weglassen, integriert @FEHLERF zwischen 0 und der *Untergrenze*.

Hinweise

@FEHLERF berechnet die Fehlerfunktion näherungsweise innerhalb von $\pm 1.2 \times 10^{-7}$.

Beispiele

@FEHLERF(0,7) = 0,677801

@FEHLERF(0,8) = 0,742101

@FEHLERF(0,7;0,8) = 0,0643, die Differenz zwischen den beiden vorherigen Beispielen.

Ähnliche @Funktionen

@FEHLERFK berechnet die komplementäre Fehlerfunktion. @FEHLERFD berechnet die Ableitung der Fehlerfunktion.

@FEHLERFK

@FEHLERFK(x) berechnet die komplementäre Fehlerfunktion, die zwischen x und ∞ integriert ist.

Argumente

x kann ein beliebiger Wert sein.

Hinweise

@FEHLERFK(x) ist gleich $1 - \text{@FEHLERF}(x)$.

@FEHLERFK berechnet die komplementäre Fehlerfunktion näherungsweise innerhalb eines Bereichs von $\pm 3 \cdot 10^{-7}$.

Beispiele

@FEHLERFK(0,7) = 0,322199

Ähnliche @Funktionen

@FEHLERF berechnet die Fehlerfunktion, die zwischen der angegebenen Ober- und Untergrenze integriert ist.

@FEHLERFD berechnet die Ableitung der Fehlerfunktion.

@FEHLERFD

@FEHLERFD(x) berechnet die Ableitung der Fehlerfunktion.

Argumente

x kann ein beliebiger Wert sein.

Beispiele

@FEHLERFD(0,7) = 0,691275

Ähnliche @Funktionen

@FEHLERF berechnet die Fehlerfunktion, die zwischen der angegebenen Ober- und Untergrenze integriert ist.

@FEHLER

@FEHLER ergibt den Wert FEHLER.

Hinweise

@FEHLER wird bei der Markierung von Fehlern in Berechnungen benutzt. Diese @Funktion wird selten alleine verwendet. Wenn @FEHLER z.B. als Argument mit @WENN benutzt wird, ergibt sich der Wert FEHLER, wenn bestimmte Bedingungen vorhanden sind, z.B. wenn eine Formel einen nicht annehmbaren Wert erzeugt (wie z.B. eine negative monatliche Zahlung).

Beispiele

@WENN(B14>3;@FEHLER;B14) = FEHLER, wenn der Wert in Zelle B14 größer als 3 ist.

Ähnliche @Funktionen

@NV ergibt den Wert NV (nicht verfügbar). @ISTFEHLER testet auf den Wert FEHLER.

@GERADE

@GERADE(x) rundet den Wert x zur nächsten geraden Ganzzahl auf.

Argumente

x ist ein beliebiger Wert. Wenn x eine gerade Ganzzahl ist, ergibt @GERADE den Wert x.

Beispiele

@GERADE(2,25) = 4

@GERADE(2) = 2

@GERADE(-2,25) = -4

Ähnliche @Funktionen

@UNGERADE rundet einen Wert zur nächsten ungeraden Ganzzahl auf. @RUNDEN, @ABRUNDEN und @AUFRUNDEN runden einen Wert auf die angegebene Anzahl von Dezimalstellen. @RUNDENM rundet einen Wert auf ein angegebenes Vielfaches. @GANZZAHL schneidet einen Wert ab, wobei die Dezimalteile ignoriert werden. @ABKÜRZEN schneidet einen Wert auf eine angegebene Dezimalstelle ab.

@GLEICH

@GLEICH(*Zeichenfolge1*;*Zeichenfolge2*) vergleicht zwei Zeichenfolgen. Wenn die beiden Zeichenfolgen genau übereinstimmen, ergibt **@GLEICH** den Wert 1 (wahr). Ansonsten ergibt **@GLEICH** den Wert 0 (falsch).

Argumente

Zeichenfolge1 und *Zeichenfolge2* stellen Text in " " (Anführungszeichen), Formeln, die Text ergeben, oder Adressen bzw. Namen von Zellen dar, die Labels oder Formeln enthalten, die Labels ergeben.

Hinweise

@GLEICH ist genauer als = (der Gleichheitszeichen-Operator) in einer Formel. Im Gegensatz zu = (dem Gleichheitszeichen-Operator) unterscheidet **@GLEICH** zwischen Groß- und Kleinbuchstaben und zwischen Buchstaben mit und ohne Akzenten.

Mit **@GLEICH** können Sie Paßwörter für Makros festlegen, indem Sie die Benutzereingabe mit der erforderlichen Eingabe vergleichen, bevor der Makro fortgesetzt wird.

Beispiele

@GLEICH("ATHEN";"Athen") = 0 (falsch).

@GLEICH("Fällig";B2) = 1 (wahr), wenn Zelle B2 das Label Fällig enthält.

@GLEICH("400";400) = FEHLER, da *Zeichenfolge2* ein Wert ist.

@EXP

@EXP(x) berechnet den Wert der Konstanten e (etwa 2,718282) in die Potenz x erhoben.

Argumente

x kann ein Wert von etwa -709.7827 bis etwa 709.7827 sein.

Hinweise

Wenn x größer als etwa 709.7827 oder kleiner als etwa -709.7827 ist, ist die Berechnung zu groß und 1-2-3 kann sie nicht speichern. In diesem Fall gibt @EXP den Wert FEHLER zurück.

Beispiele

@EXP(0,7) = 2,013753

Ähnliche @Funktionen

@EXP2 berechnet den Wert von e in die Potenz von x^2 erhoben. @LN ist die Umkehrfunktion von @EXP.

@EXP2

@EXP2(x) berechnet den Wert der Konstanten e (ca. 2,718282) in die Potenz von (x²) erhoben.

Argumente

x ist ein Wert von ca. -106,570 bis ca. 106,570.

Hinweise

Wenn x größer als etwa 106,570 oder kleiner als etwa -106,570 ist, ist die Berechnung zu groß, und 1-2-3 kann sie nicht speichern. In diesem Fall ergibt @EXP2 den Wert FEHLER. Wenn x kleiner als ca. 15,098 oder größer als ca. 15,098 ist, kann 1-2-3 den Wert von @EXP2 berechnen und speichern, kann ihn jedoch nicht anzeigen. (In der Zelle wird eine Reihe von Sternchen angezeigt.)

Beispiele

@EXP2(0,7) = 0,612626

Ähnliche @Funktionen

@EXP berechnet e, erhöht in eine bestimmte Potenz.

@FAKULTÄT

@FAKULTÄT(n) berechnet den Faktor von n .

Argumente

n ist eine beliebige positive Ganzzahl oder 0.

Hinweise

Der Faktor von n ist gleich dem Produkt aller positiven Ganzzahlen von 1 bis n .

Wenn n größer als oder gleich 171 ist, ist die Berechnung zu groß, und 1-2-3 kann sie nicht speichern. In diesem Fall ergibt @FAKULTÄT den Wert FEHLER.

Beispiele

@FAKULTÄT(0) = 1

@FAKULTÄT(5) = 120, das Ergebnis von $1*2*3*4*5$.

Ähnliche @Funktionen

@FAKULTÄTLN berechnet den natürlichen Logarithmus des Faktors von n @PRODUKT multipliziert die Werte in einer Liste.

@FAKULTÄTLN

@FAKULTÄTLN(n) berechnet den natürlichen Logarithmus des Faktors von n .

Argumente

n ist eine beliebige positive Ganzzahl oder 0.

Hinweise

Der Faktor von n ist gleich dem Produkt aller positiven Ganzzahlen von 1 bis n .

Beispiele

@FAKULTÄTLN(0) = 0, das Ergebnis von $\ln(1)$.

@FAKULTÄTLN(5) = 4,787492, das Ergebnis von $\ln(1*2*3*4*5)$.

Ähnliche @Funktionen

@FAKULTÄT berechnet den Faktor von n . @LN berechnet den natürlichen Logarithmus eines Wertes.

@FALSCH

@FALSCH gibt den logischen Wert 0 (falsch) zurück.

Hinweise

Wenn eine logische Anweisung wie z.B. $A1=B1$ wahr ist, ist der logische Wert 1. Ist sie falsch, so ist der logische Wert 0.

Die Benutzung von @FALSCH ist gleichbedeutend mit der Benutzung des Wertes 0 in Formeln, mit denen logische Bedingungen ausgewertet werden. Mit @FALSCH wird die Formel jedoch verständlicher.

Benutzen Sie @FALSCH mit Makros oder @Funktionen, wie z.B. @WENN und @WAHL, die einen logischen Wert von 0 (falsch) erfordern. @FALSCH kann auch als y-Argument für @WENN benutzt werden, d.h. für den Wert, der zurückgegeben wird, wenn die Bedingung nicht erfüllt ist.

Beispiele

@WENN(A6>500;@WAHR;@FALSCH) = 0, wenn Zelle A6 einen Wert enthält, der kleiner als oder gleich 500 ist.

Ähnliche @Funktionen

@WAHR gibt den logischen Wert 1 wieder.

@FVERTEILUNG

@FVERTEILUNG(x;Freiheitsgrad1;Freiheitsgrad2;[Typ]) berechnet die F-Verteilung.

Argumente

x ist der Wert, mit dem die F-Verteilung ausgewertet werden soll. Der Wert, den Sie für x eingeben, hängt von dem Wert ab, den Sie für Typ eingeben.

| <u>Typ</u> | <u>x</u> |
|------------|--|
| 0 | Der kritische Wert oder obere Grenzwert für den Wert der kumulativen F-Verteilung; entspricht einem Wert größer als oder gleich 0; Vorgabe, wenn Sie kein Argument angeben |
| 1 | Eine Wahrscheinlichkeit und ein Wert von 0 bis 1 |

Freiheitsgrad1 und Freiheitsgrad2 entsprechen den Zahlen für den Freiheitsgrad der ersten bzw. zweiten Stichprobe. Freiheitsgrad1 und Freiheitsgrad2 sind positive Ganzzahlen.

Typ ist ein wahlweises Argument, mit dem angegeben wird, wie 1-2-3 @FVERTEILUNG berechnet.

| <u>Typ</u> | <u>1-2-3 berechnet</u> |
|------------|--|
| 0 | Die Signifikanzebene, die dem kritischen Wert x entspricht; Vorgabe, wenn Sie das Argument weglassen |
| 1 | Den kritischen Wert, der der Signifikanzebene x entspricht |

Hinweise

@FVERTEILUNG berechnet die kumulative F-Verteilung näherungsweise innerhalb eines Bereichs von $\pm 3 \cdot 10^{-7}$. Wenn @FVERTEILUNG das Ergebnis nach 100 Berechnungsversuchen nicht näherungsweise mit einer Genauigkeit von 0,0000001 berechnen kann, ist das Ergebnis FEHLER.

Die F-Verteilung ist eine fortlaufende Verteilung, die sich aus der Beziehung der beiden Chi-Quadrat-Verteilungen ergibt, die jeweils durch die Anzahl von Freiheitsgraden dividiert werden.

Mit @FVERTEILUNG bestimmen Sie den Grad, um den zwei Stichproben variieren.

Beispiele

@FVERTEILUNG(3,07;8;10) = 0,05

@FVERTEILUNG(0,05;8;10) = 0.999865

Ähnliche @Funktionen

@FTEST berechnet die Wahrscheinlichkeit, die mit einem F-Test verbunden ist. @CHIVERTEILUNG berechnet die Chi-Quadrat-Verteilung. @TVERTEILUNG berechnet die Studentsche t-Verteilung.

@FINDEN

@FINDEN(*Suchfolge*; *Zeichenfolge*; *Startnummer*) berechnet die Position in *Zeichenfolge*, bei der *Suchfolge* das erste Mal auftritt. **@FINDEN** durchsucht *Zeichenfolge* ab der Position, die mit *Startnummer* angegeben wird.

Argumente

Suchfolge und *Zeichenfolge* entsprechen Text in " " (Anführungszeichen), einer Formel, die eine Zeichenfolge ergibt, oder der Adresse bzw. dem Namen einer Zelle, die *Zeichenfolge* enthält, bzw. einer Formel, die ein Label ergibt.

Startnummer ist ein Zählwert.

Hinweise

Wenn 1-2-3 die *Suchfolge* in *Zeichenfolge* nicht findet, ergibt **@FINDEN** den Wert FEHLER. **@FINDEN** gibt außerdem FEHLER zurück, wenn *Startnummer* größer als die Anzahl von Zeichen in *Zeichenfolge* ist oder wenn *Startnummer* negativ ist.

@FINDEN unterscheidet zwischen Groß- und Kleinbuchstaben und zwischen Buchstaben mit und ohne Akzentzeichen; so findet **@FINDEN** z.B. die *Suchfolge* "e" in der *Zeichenfolge* "EINE." nicht.

@FINDEN wird auch in Verbindung mit @MITTE oder @ERSETZEN benutzt, um Zeichenfolgen zu suchen und zu extrahieren bzw. zu ersetzen.

Beispiele

@FINDEN("P";"Offene Posten";0) = 8, da die *Suchfolge* P an Stelle 8 in der *Zeichenfolge* Offene Posten steht.

@ZUKWERT

@ZWERT

@ZUKWERT(*Zahlungen*; *Zinssatz*; *Laufzeit*) berechnet den zukünftigen Wert einer Anlage auf der Grundlage von gleichen *Zahlungen* mit einem periodischen *Zinssatz* über die *Laufzeit*.

@ZWERT(*Zahlungen*; *Zinssatz*; *Laufzeit*; [*Typ*]; [*Aktwert*]) berechnet den zukünftigen Wert einer Anlage mit einem bestimmten *Aktwert* auf der Grundlage einer Reihe von gleichen *Zahlungen* einem periodischen *Zinssatz* über die *Laufzeit*. @ZWERT nimmt die Berechnung entweder für eine nachschüssige oder eine vorschüssige Rente vor, je nach Wert, den Sie für *Laufzeit* angeben.

Argumente

Zahlungen und *Laufzeit* sind Werte.

Für @ZWERT muß *Laufzeit* ein positiver Wert sein.

Zinssatz ist ein größerer Wert als -1.

Typ ist ein wahlweises Argument, mit dem angegeben wird, ob für eine nachschüssige Rente oder eine vorschüssige Rente berechnet werden soll.

| <u>Typ</u> | <u>1-2-3 berechnet für</u> |
|------------|---|
| 0 | Nachschüssige Rente (Zahlungen sind am Ende der Periode fällig); Vorgabe, wenn Sie das Argument weglassen |
| 1 | Vorschüssige Rente (Zahlung am Anfang einer Periode fällig) |

Aktwert ist ein wahlweises Argument, mit dem der aktuelle Wert einer Reihe von zukünftigen Zahlungen angegeben wird. *Aktwert* kann ein beliebiger Wert sein. Wenn Sie das Argument *Aktwert* weglassen, benutzt 1-2-3 den Wert 0.

Sie können ein wahlweises Argument nicht ohne die davorstehenden wahlweisen Argumente benutzen.

Hinweise

Der Zeitraum zur Berechnung der *Zinsen* muß derselbe Zeitraum wie für *Laufzeit* sein; wenn Sie z.B. eine monatliche Zahlung berechnen, geben Sie *Zinssatz* und *Perioden* in monatlichen Schritten ein. Im allgemeinen bedeutet dies, daß Sie den *Zinssatz* durch 12 dividieren und die Anzahl von Jahren in *Laufzeit* mit 12 multiplizieren müssen.

Beispiele

Sie können jedes Jahr für die Dauer von 20 Jahren 2.000 DM jährlich auf ein Konto einzahlen, um auf eine Rente zu sparen. Für das Konto werden jährlich 7,5% Zinsen gutgeschrieben. Die Zinsen werden am letzten Tag jedes Jahres bezahlt. Nun möchten Sie den Stand Ihres Kontos in 20 Jahren berechnen. Die Einzahlung in den einzelnen Jahren erfolgt am letzten Tag des Jahres.

@ZUKWERT(2000;0,075;20) = 86.609,36 DM, der Betrag auf Ihrem Konto nach Ablauf von 20 Jahren.

Wenn Sie die jährliche Einzahlung am ersten Tag des Jahres vornehmen:

@ZWERT(2000;0,075;20;1) = 93.105,06 DM

Ähnliche @Funktionen

@ZUKWERT2 berechnet den zukünftigen Wert einer Kapitalanlage, basierend auf einer Reihe von gleichen Zahlungen und einem gleichbleibenden *Zinssatz* über die Anzahl der Zahlungsperioden in der *Laufzeit* bei vorschüssiger Rente. @ZWERTBETRAG berechnet den zukünftigen Wert eines bei einem bestimmten *Zinssatz* angelegten Pauschalbetrags für eine gegebene Anzahl von *Perioden*.

@AKTWERT und @AWERT bestimmen den aktuellen Wert einer Anlage. @NETAKTWERT berechnet den aktuellen Nettowert einer Investition, wobei der künftige Wert auf den aktuellen Wert diskontiert wird.

Beispiel: @INDEX

In einer Tabelle namens ERHÖHUNG (A3..E8) werden die Gehaltserhöhungen je nach Bewertung der Mitarbeiter aufgeführt.

@INDEX(ERHÖHUNG;2;3) in eine als Prozent formatierte Zelle eingegeben, ergibt 5%, die Gehaltserhöhung für einen Mitarbeiter mit der Bewertung 3 und der Gehaltskategorie 2.

@INDEX(ERHÖHUNG;1;2) in eine als Prozent formatierte Zelle eingegeben, ergibt 7%, die Gehaltserhöhung für einen Mitarbeiter mit einer Bewertung 2 und einer Gehaltskategorie 1.

| A | B | C | D | E |
|---|---|---|---|---|
| 1 | | | | |
| 2 | | | | |
| 3 | | | | |
| 4 | | | | |
| 5 | | | | |
| 6 | | | | |
| 7 | | | | |
| 8 | | | | |

| | -- G E H A L T S K A T E G O R I E -- | | | |
|-----------|---------------------------------------|----|----|----|
| Bewertung | 1 | 2 | 3 | 4 |
| 1 | 10% | 9% | 8% | 7% |
| 2 | 7% | 6% | 5% | 4% |
| 3 | 6% | 5% | 4% | 3% |
| 4 | 3% | 2% | 1% | 0% |
| 5 | 0% | 0% | 0% | 0% |

@GAMMA

@GAMMA(x) berechnet die Gamma-Funktion.

Argumente

x ist ein beliebiger Wert mit Ausnahme von 0 und negativen Ganzzahlen.

Hinweise

@GAMMA berechnet die Gammaverteilung genau innerhalb von sechs signifikanten Zahlen.

Beispiele

@GAMMA(0,5) = 1,772454

@GAMMA(5) = 24

Ähnliche @Funktionen

@BETA berechnet die Beta-Funktion. @GAMMAI berechnet die unvollständige Gamma-Funktion. @GAMMALN berechnet den natürlichen Logarithmus der Gamma-Funktion.

@GAMMAI

@GAMMAI($a;x;[Komplement]$) berechnet die unvollständige Gamma-Funktion.

Argumente

a ist ein positiver Wert.

x ist ein positiver Wert oder 0.

Komplement ist ein wahlweises Argument, mit dem angegeben wird, wie 1-2-3 @GAMMAI berechnet.

Komplemen 1-2-3 berechnet

t

0 P($a;x$); Vorgabe, wenn Sie das Argument weglassen

1 Q($a;x$); entspricht 1-P($a;x$)

Hinweise

@GAMMAI berechnet die unvollständige Gamma-Funktion genau innerhalb von sechs signifikanten Ziffern.

Beispiele

@GAMMAI(7,5;12,497;1) = 0,050024

Ähnliche @Funktionen

@GAMMA berechnet die Gamma-Verteilungsfunktion. @GAMMALN berechnet den natürlichen Logarithmus der Gamma-Funktion.

@GAMMALN

@GAMMALN(x) berechnet den natürlichen Logarithmus der Gamma-Funktion.

Argumente

x ist ein beliebiger Wert größer als 0.

Hinweise

@GAMMALN berechnet den natürlichen Logarithmus der Gamma-Funktion genau innerhalb von sechs signifikanten Zahlen.

Beispiele

@GAMMALN(0,5) = 0,572365

@GAMMALN(5) = 3,178054

Ähnliche @Funktionen

@GAMMA berechnet die Gamma-Verteilungsfunktion. @GAMMAI berechnet die unvollständige Gamma-Funktion.

@GEOMITTEL

@GEOMITTEL(*Argument-Liste*) berechnet das geometrische Mittel der Werte in *Argument-Liste*.

Argumente

Argument-Liste kann jede der folgenden Angaben in einer beliebigen Kombination enthalten: Zahlen, numerische Formeln, deren Auswertung Zahlen ergibt, sowie Adressen oder Namen von Bereichen, die Zahlen oder numerische Formeln enthalten. Trennen Sie die einzelnen Elemente in *Argument-Liste* durch Argument-Trennzeichen.

Alle Werte in *Argument-Liste* müssen größer als 0 sein.

Siehe auch Argumente für statistische @Funktionen.

Hinweise

Das geometrische Mittel einer *Argument-Liste* mit n Werten entspricht der n -ten Wurzel des Produktes der Werte in *Argument-Liste*.

Bei derselben *Argument-Liste* ist das Ergebnis von @GEOMITTEL kleiner als das Ergebnis von @MITTELWERT, es sei denn, alle Werte in *Argument-Liste* sind gleich. Wenn alle Werte in *Argument-Liste* gleich sind, sind die Ergebnisse von @GEOMITTEL und @MITTELWERT gleich.

Beispiele

@GEOMITTEL(A1..A4) = 239,1886, wenn A1..A4 die Werte 160, 227, 397 und 227 enthält.

Ähnliche @Funktionen

@HARMITTEL berechnet das harmonische Mittel der Werte in einer *Argument-Liste*. @MITTELWERT und @PURMITTELWERT berechnen den Mittelwert der Werte in einer *Argument-Liste*. @MEDIAN berechnet den Medianwert in einer Liste mit Werten.

@HARMITTEL

@HARMITTEL(*Argument-Liste*) berechnet das harmonische Mittel der Werte in *Argument-Liste*.

Argumente

Argument-Liste kann eine der folgenden Angaben in einer beliebigen Kombination enthalten: Zahlen, numerische Formeln, deren Auswertung Zahlen ergibt, sowie Adressen und Namen von Bereichen, die Zahlen oder numerische Formeln enthalten. Trennen Sie die Elemente in *Argument-Liste* durch Argument-Trennzeichen.

Alle Werte in *Argument-Liste* müssen größer als 0 sein.

Siehe auch Argumente für statistische @Funktionen.

Hinweise

Das harmonische Mittel der Werte in *Argument-Liste* ist der Umkehrwert des arithmetischen Mittels der Umkehrwerte der Werte in *Argument-Liste*. So ist z.B. der Mittelwert von 1/2, 1/3, und 1/4 gleich 13/36; deshalb ist das harmonische Mittel von 2, 3 und 4 $36/13$.

Bei derselben *Argument-Liste* ist das Ergebnis von @HARMITTEL immer kleiner als das Ergebnis von @GEOMITTEL.

Beispiele

@HARMITTEL(25;50;75) = 40,90909

Ähnliche @Funktionen

@GEOMITTEL berechnet das geometrische Mittel der Werte in einer *Argument-Liste*. @MITTELWERT und @PURMITTELWERT berechnen den Mittelwert der Werte in einer *Argument-Liste*. @MEDIAN berechnet den Medianwert in einer Liste mit Werten.

@HEX

@HEX(x) wandelt eine Dezimalzahl in das Hexadezimal-Äquivalent um.

Argumente

x ist eine Ganzzahl von -2.147.483.648 bis 2.147.483.647. Wenn x keine Ganzzahl ist, schneidet 1-2-3 sie auf eine Ganzzahl ab.

Hinweise

Hexadezimalwerte von 00000000 bis 7FFFFFFF entsprechen 0 und positiven Dezimalwerten.

Hexadezimalwerte von 80000000 bis FFFFFFFF entsprechen negativen Dezimalwerten.

Beispiele

@HEX(162) = A2

Ähnliche @Funktionen

@DEZIMAL wandelt Hexadezimalzahlen in Dezimalzahlen um.

@HVERWEIS

@HVERWEIS(*Argument*; *Bereich*; *Versatz*) ermittelt den Inhalt einer Zelle in einer bestimmten Zeile einer horizontalen Verweistabelle, d.h. einen Bereich mit Wertinformationen in steigender Folge oder Labels in der ersten Zeile.

Argumente

Argument ist ein Wert oder Text je nach Inhalt der ersten Zeile der horizontalen Verweistabelle.

| Erste Reihe | x |
|-------------|---|
| Werte | Ein beliebiger Wert, der größer als oder gleich dem ersten Wert in <i>Bereich</i> ist. Ist <i>Argument</i> kleiner als der erste Wert in <i>Bereich</i> , so ergibt @HVERWEIS FEHLER. Ist <i>Argument</i> größer als der letzte Wert in der ersten Zeile von <i>Bereich</i> , so stoppt @HVERWEIS bei der letzten Zelle in der Zeile, die mit <i>Versatz</i> angegeben wird, und gibt den Inhalt dieser Zelle als Antwort zurück. |
| Labels | Text in " " (Anführungszeichen), eine Formel, die Text ergibt, oder die Adresse bzw. den Namen einer Zelle, die ein Label oder eine Formel enthält, die ein Label ergibt. Entspricht <i>Argument</i> nicht genau dem Inhalt einer Zelle in der ersten Zeile von <i>Bereich</i> , so ergibt @HVERWEIS FEHLER. |

Bereich entspricht der Position der horizontalen Verweistabelle. Bei *Bereich* kann es sich um eine beliebige Bereichsadresse oder einen Bereichsnamen handeln. Ist *Bereich* ein 3D-Bereich, so benutzt 1-2-3 nur das erste Arbeitsblatt in *Bereich*.

Versatz ist ein Zählwert, der der Position entspricht, die die Zeile in *Bereich* belegt.

Hinweise

@HVERWEIS vergleicht *Argument* mit jeder Zelle in der ersten Zeile der Tabelle. Wenn 1-2-3 eine Zelle in der ersten Zeile findet, die *Argument* enthält (oder wenn *Argument* ein Wert ist, den Wert, der *Argument* am nächsten kommt, jedoch nicht größer ist als *Argument*), geht es in dieser Spalte um die Anzahl von Zeilen nach unten, die mit *Versatz* angegeben wird, und gibt den Inhalt dieser Zelle als Antwort zurück.

Beispiele

@HVERWEIS

Ähnliche @Funktionen

@VVERWEIS ermittelt den Inhalt einer Zelle in einer vertikalen Verweistabelle. @INDEX ermittelt den Inhalt einer Zelle, wenn Sie Zählwerte für Spalte und Zeile angeben. @WAHL ersetzt eine Verweistabelle, für die nur eine Zeile erforderlich ist. @VERGLEICH ermittelt die relative Position einer Zelle mit dem angegebenen Inhalt.

@XINDEX ermittelt den Inhalt einer Zelle, die mit Spalten-, Zeilen- und Arbeitsblatt-Titeln gekennzeichnet wird.

@MAXVERWEIS ergibt eine absolute Referenz auf die Zelle, die den größten Wert in einer Liste mit Bereichen enthält. @MINVERWEIS ergibt eine absolute Referenz auf die Zelle, die den kleinsten Wert in einer Liste mit Bereichen enthält.

@STUNDE

@STUNDE(*Zeitseriennummer*) berechnet die Stunde, d.h. einen Wert zwischen 0 (Mitternacht) und 23 (23:00 oder 11:00 PM) aus einer *Zeitseriennummer*.

Argumente

Zeitseriennummer ist ein Wert von 0,000000 (Mitternacht) bis 0,999988 (23:59:59). Im allgemeinen wird die *Zeitseriennummer* von einer anderen Zeit-@Funktion geliefert.

Hinweise

Der Stundenanteil wird besonders in Berechnungen benutzt, bei denen ganze Stunden verwendet werden, wie z.B. bei der Berechnung von Stundenlöhnen oder Stunden seit Beginn der Arbeit an einem bestimmten Projekt. Außerdem können Sie mit dieser @Funktion auch ein Arbeitsblatt mit einem Zeitstempel versehen.

Beispiele

@STUNDE(0,51565) = 12, da 0,51565 die *Zeitseriennummer* für 12:22:32 ist.

@STUNDE(**@ZEIT**(13;45;18)) = 13 (1:00 PM), da 13 das *Stunde*-Argument für **@ZEIT**(13;45;18) ist.

Ähnliche @Funktionen

@MINUTE berechnet die Minuten und **@SEKUNDE** die Sekunden aus einer *Zeitseriennummer*.

@WENN

@WENN(*Bedingung*;x;y) wertet *Bedingung* aus und gibt je nach Ergebnis der Auswertung einen von zwei Werten zurück. Wenn *Bedingung* wahr ist, gibt **@WENN** x zurück; ist *Bedingung* falsch, so gibt **@WENN** den Wert y zurück.

Argumente

Bedingung ist im allgemeinen eine logische Formel. Sie können jedoch jede beliebige Formel, Zahl, Text in " " (Anführungszeichen) oder einen Namen bzw. die Adresse einer Zelle als *Bedingung* benutzen. 1-2-3 wertet jede *Bedingung*, die nicht gleich null ist, als wahr, und jede *Bedingung*, die gleich null ist, als falsch aus. Leere Zellen, Text und FEHLER sowie NV entsprechen alle null, wenn sie als *Bedingung* benutzt werden.

x und y sind Werte, Text in " " (Anführungszeichen) oder die Adressen bzw. Namen von Zellen, die Werte oder Labels enthalten.

Hinweise

@WENN wird besonders in Verbindung mit @FEHLER und @NV benutzt, um Fehler oder fehlende Daten in Formeln zu kennzeichnen. Außerdem wird diese @Funktion benutzt, um FEHLER, NV und Rechenfehler in Fällen zu umgehen, in denen Daten fehlen oder ungenau sind, z.B. um eine Division durch null zu verhindern.

Sie können **@WENN** Funktionen ineinander verschachteln, um eine komplexe Bedingung zu schaffen. So verschachtelt z.B.

@WENN(UMSATZ>10000;UMSATZ*0,15;**@WENN**(UMSATZ>5000;UMSATZ*0,10;UMSATZ*0,02))

zwei **@WENN** Funktionen, um eine Provision auf drei Umsatzebenen zu ermitteln: Gesamtumsatz größer als 10.000 DM, Gesamtumsatz größer als 5.000 DM und Gesamtumsatz kleiner als oder gleich 5.000 DM.

Beispiele

@WENN(SALDO>=0;SALDO;"Überzogen") gibt den Wert in der Zelle SALDO zurück, wenn der Wert in SALDO 0 oder positiv ist, bzw. gibt das Label Überzogen zurück, wenn der Wert in SALDO negativ ist.

@INDEX

@INDEX(*Bereich*; *Spalte*; *Zeile*; [*Arbeitsblatt*]) ermittelt den Inhalt der Zelle, die in der angegebenen *Spalte*, *Zeile* und (wahlweise) in dem angegebenen *Arbeitsblatt* eines *Bereichs* steht.

Argumente

Bereich ist eine Bereichsadresse oder ein Bereichsname.

Spalte ist die Nummer der Spalte, die **@INDEX** benutzt.

Zeile ist die Nummer der Zeile, die **@INDEX** benutzt, bzw. die Adresse oder der Name einer Zelle, die 0 oder eine positive Ganzzahl enthält.

Arbeitsblatt ist ein wahlweises Argument, das der Nummer des Arbeitsblattes entspricht, das **@INDEX** benutzt. Wenn Sie kein *Arbeitsblatt* angeben, benutzt **@INDEX** nur das erste Arbeitsblatt in Bereich.

Beispiele

@INDEX

Ähnliche @Funktionen

@HVERWEIS und **@VVERWEIS** ermitteln Einträge in horizontalen und vertikalen Verweistabellen. **@WAHL** ermittelt einen Eintrag in einer Liste.

@VERGLEICH ermittelt die relative Position einer Zelle mit dem angegebenen Inhalt. **@XINDEX** ermittelt den Inhalt einer Zelle, die mit Spalten-, Zeilen und Arbeitsblatt-Titeln angegeben wird. **@MAXVERWEIS** ergibt eine absolute Referenz auf die Zelle, die den größten Wert in einer Liste mit Bereichen enthält. **@MINVERWEIS** ergibt eine absolute Referenz auf die Zelle, die den kleinsten Wert in einer Liste mit Bereichen enthält.

@INFO

@INFO(*Attribut*) gibt Informationen über die laufende Arbeit mit 1-2-3 zurück.

Argumente

Attribut ist eines der folgenden Elemente, als Text in " " (Anführungszeichen) eingegeben, bzw. die Adresse oder der Name einer Zelle, die eines der folgenden Elemente enthält:

| Attribut | Ergebnis |
|-----------------------|--|
| anzdatei | Die Anzahl von gerade geöffneten Arbeitsmappen |
| arbeitsblätter-anzahl | Die Anzahl der Blätter in der aktuellen Arbeitsmappe |
| auswahl | Die Adresse des gerade ausgewählten Bereichs oder der Name des gerade ausgewählten Diagramms, des Zeichenobjekts oder der Abfragetabelle |
| auswahl-teil | NV |
| auswahl-typ | Der gerade ausgewählte Typ: Bereich, Zeichenobjekt, Abfrage, Diagramm |
| autor | Der Benutzername der Person, die die aktuelle Arbeitsmappe zuerst gespeichert hat |
| bearbeitungszeit | Eine Zeitseriennummer, die der Gesamtanzahl von Stunden und Minuten entspricht, die die aktuelle Arbeitsmappe geöffnet war |
| benutzervorgaben-name | Ihr Email- oder Netzwerk-Benutzername |
| bildschirm-breite | Die Breite des Bildschirms in Pixel |
| bildschirm-höhe | Die Höhe des Bildschirms in Pixel |
| bsreturncode | NV |
| bssystem | Der Name des Betriebssystems |
| bsversion | NV |
| dateiumfang | Die Größe der aktuellen Arbeitsmappe in Kilobyte (KB) |
| dbdatensatzzahl | NV |
| dbreturncode | NV |
| dbtreibermeldung | NV |
| erstell-datum | Eine Datumseriennummer, die dem Datum entspricht, an dem die aktuelle Arbeitsmappe zuerst gespeichert wurde |
| geändert-von | Der Benutzername der Person, die die aktuelle Arbeitsmappe zuletzt gespeichert hat |
| gesamtspcich | Der gesamte zur Verfügung stehende Speicherplatz (sowohl der momentan zur Verfügung stehende als auch der schon belegte Speicherplatz) |
| letzte-änderung | Eine Datumseriennummer, die dem Datum entspricht, an dem die aktuelle Arbeitsmappe zuletzt gespeichert wurde |
| makro-fehlersuche | Ja bei geöffnetem Fehlersuchefenster; Nein bei geschlossenem |
| makro-schritt | Ja bei eingeschaltetem Schritt-Modus; |

| | |
|--------------|--|
| | Nein bei ausgeschaltetem |
| modus | Der aktuelle Modus: 0 WARTEN 1 BEREIT 2 LABEL 3 MENÜ 4 WERT 5 ZEIGEN 6 EDIT 7 FEHLER 8 FINDEN 9 DATEIEN 10 HILFE 11 STAT 13 NAMEN 99 Alle anderen Modi (z.B. vom Benutzer mit {ANZEIGE} definiert) |
| neuberech | Die aktuelle Neuberechnungs-Methode in Form der Zeichenfolge "Automatisch" oder "Manuell" |
| ursprung | Die Adresse der ersten Zelle in dem Fenster, in dem der Zellzeiger steht |
| verfügspeich | Der zur Verfügung stehende Speicherplatz |
| version | Die Versionsnummer des gerade benutzten 1-2-3 für Windows Produktes |
| verzeichnis | Der aktuelle <u>Pfad</u> , einschließlich dem Laufwerkbuchstaben |
| winverz | Der Pfad zum Windows-Verzeichnis, einschließlich dem Laufwerkbuchstaben |

Hinweise

Zusätzlich zu den hier aufgelisteten Attributen kann als *Attribut* auch ein beliebiges Infoelement verwendet werden. Berechnen Sie die Daten neu (indem Sie F9 (KALK) drücken), bevor Sie @INFO verwenden, damit die Ergebnisse richtig sind.

Benutzen Sie @INFO in Makros, wenn Sie dem Benutzer oder dem Makro Informationen über den Status von 1-2-3 liefern müssen (z.B. dem Benutzer den aktuellen Pfad in einem Makro angeben müssen, mit dem Arbeitsmappen automatisch gespeichert werden, oder um den Benutzer zu warnen, wenn der Speicherplatz langsam zu Ende geht).

Verwenden Sie @INFO mit @WENN zur Überprüfung des Status von 1-2-3 und um einem Makro mitzuteilen, was bei bestimmten Bedingungen geschehen soll. So kann er z.B. den Pfad wechseln oder nicht mehr benötigte Daten löschen bzw. geöffnete Arbeitsmappen schließen, wenn der Speicherplatz knapp wird.

Beispiele

@INFO("anzdatei") = 2, wenn zwei Arbeitsmappen geöffnet sind.

@INFO(B4) = 3, wenn B4 das Label Modus enthält und 1-2-3 sich im MENÜ-Modus befindet.

Ähnliche @Funktionen

@ZELLE gibt Informationen über eine bestimmte Zelle an. @ZELLZEIGER gibt Informationen über die aktuelle Zelle an.

@GANZZAHL

@GANZZAHL(x) gibt den ganzzahligen Teil von x zurück.

Argumente

x ist ein Wert.

Hinweise

Verwenden Sie das feste Zahlenformat, um die Werte mit einer bestimmten Anzahl von Dezimalstellen anzuzeigen, wenn 1-2-3 Werte mit voller Genauigkeit berechnen soll. In diesem Fall benutzen Sie @GANZZAHL nicht.

Beispiele

@GANZZAHL(35,67) = 35.

@GANZZAHL(@JETZT) = die Datumseriennummer für das laufende Datum und die laufende Uhrzeit, da der Zeitteil ein Dezimalwert ist.

Ähnliche @Funktionen

@RUNDEN, @AUFRUNDEN und @ABRUNDEN runden einen Wert auf das nächste Vielfache der angegebenen Zehnerpotenz. @RUNDENM rundet einen Wert auf ein angegebenes Vielfaches. @GERADE rundet einen Wert nach oben auf die nächste gerade Ganzzahl. @UNGERADE rundet einen Wert nach oben auf die nächste ungerade Ganzzahl. @ABKÜRZEN schneidet einen Wert auf eine angegebene Anzahl von Dezimalstellen ab.

@ZINSBETRAG

@TILGUNG

@ZINSBETRAG(*Kapital*; *Zinssatz*; *Laufzeit*; *Anfangsperiode*; [*Endperiode*]; [*Typ*]; [*Zukwert*]) berechnet den kumulativen Zinsanteil der periodischen Zahlung eines Darlehens (*Kapital*) zu einem bestimmten *Zinssatz* für eine bestimmte Anzahl von Zahlungsperioden (*Laufzeit*).

@TILGUNG(*Kapital*; *Zinssatz*; *Laufzeit*; *Anfangsperiode*; [*Endperiode*]; [*Typ*]; [*Zukwert*]) berechnet den Kapitalanteil der periodischen Zahlung eines Darlehens (*Kapital*) zu einem bestimmten *Zinssatz* für eine bestimmte Anzahl von Zahlungsperioden (*Laufzeit*).

Argumente

Kapital und *Laufzeit* sind Werte. *Laufzeit* kann ein beliebiger Wert mit Ausnahme von 0 sein.

Zinssatz ist ein Dezimal- oder Prozentwert größer als -1.

Anfangsperiode ist der Punkt in der Laufzeit des Darlehens, an dem Sie mit der Berechnung der Zinsen oder des Kapitals beginnen möchten. *Anfangsperiode* kann ein beliebiger Wert größer als oder gleich 1 sein, darf jedoch nicht größer sein als *Laufzeit*.

Endperiode ist der Punkt in der Laufzeit des Darlehens, an dem Sie mit der Berechnung der Zinsen oder des Kapitals aufhören möchten. *Endperiode* kann ein beliebiger Wert größer als *Anfangsperiode* sein. Wenn Sie das Argument *Endperiode* weglassen, ist *Endperiode* gleich *Anfangsperiode*.

Typ ist ein wahlweises Argument, mit dem angegeben wird, ob eine nachschüssige oder vorschüssige Rente berechnet werden soll.

Typ 1-2-3 berechnet

- | | |
|---|--|
| 0 | Nachschüssige Rente (Zahlungen erfolgen am Ende einer Periode); Vorgabe, wenn Sie das Argument weglassen |
| 1 | Vorschüssige Rente (Zahlungen erfolgen am Anfang einer Periode) |

Zukwert ist ein wahlweises Argument, mit dem der zukünftige Wert einer Reihe von Zahlungen angegeben wird. *Zukwert* kann ein beliebiger Wert sein. Wenn Sie das *Zukwert*-Argument weglassen, benutzt 1-2-3 den Wert 0.

Sie können kein wahlweises Argument ohne die davorstehenden wahlweisen Argumente benutzen.

Hinweise

Der Zeitraum, der zur Berechnung des *Zinssatzes* benutzt wird, muß identisch mit dem Zeitraum für *Laufzeit* sein; wenn Sie z.B. eine monatliche Zahlung berechnen, müssen Sie *Zinssatz* und *Laufzeit* in Monaten eingeben. Im allgemeinen bedeutet dies, daß Sie den *Zinssatz* durch 12 dividieren und die Anzahl von Jahren in *Laufzeit* mit 12 multiplizieren müssen.

Beispiele

Sie haben ein Darlehen von 8.000 DM mit einer Laufzeit von 3 Jahren und einem jährlichen *Zinssatz* von 10,5% aufgenommen, die monatlich bezahlt werden müssen. Die monatlichen Zahlungen belaufen sich auf 260,02 DM. Um den Zinsanteil der Zahlungen des letzten Jahres zu ermitteln, benutzen Sie folgende @Funktion:

@ZINSBETRAG(8000;0,105/12;36;25;36) = 170,45 DM

Um den Kapitalanteil der letztjährigen Zahlungen zu ermitteln, benutzen Sie folgende @Funktion:

@TILGUNG(8000;0,105/12;36;25;36) = 2.949,79 DM

Ähnliche @Funktionen

@RATE berechnet die periodische Zahlung für ein Darlehen.

@EFFBELA

@EFFBELA(Laufzeit;Rate;Aktwert;[Typ];[Zukwert];[Schätzwert]) berechnet den periodischen Zinssatz, der erforderlich ist, damit eine Investition (Aktwert) auf einen zukünftigen Wert Zukwert über eine Anzahl von Perioden in Laufzeit anwächst.

Argumente

Laufzeit ist eine positive Ganzzahl.

Rate und Aktwert sind Werte.

Typ ist ein wahlweises Argument, mit dem angegeben wird, ob eine nachschüssige oder vorschüssige Rente berechnet wird.

| Typ | 1-2-3 berechnet |
|-----|--|
| 0 | Nachschüssige Rente (Zahlungen erfolgen am Ende einer Periode); Vorgabe, wenn Sie das Argument weglassen |
| 1 | Vorschüssige Rente (Zahlungen erfolgen am Anfang einer Periode) |

Zukwert ist ein wahlweises Argument, mit dem der zukünftige Wert einer Reihe von Zahlungen angegeben wird. Zukwert kann ein beliebiger Wert sein. Wenn Sie das Argument Zukwert weglassen, benutzt 1-2-3 0.

Schätzwert ist ein wahlweises Argument, das Ihrer Schätzung des Zinssatzes entspricht. Schätzwert ist ein Wert von 0 bis 1. Wenn Sie das Argument Schätzwert weglassen, benutzt 1-2-3 0,10 (10%).

Sie können ein wahlweises Argument nicht ohne die davorstehenden wahlweisen Argumente benutzen.

Hinweise

@EFFBELA benutzt eine Reihe von Näherungen beginnend mit Ihrem Schätzwert, um den Zinssatz zu berechnen. Beginnen Sie mit einem Schätzwert, den Sie als für den Zinssatz angemessen betrachten. Es ist mehr als eine Lösung möglich, deshalb versuchen Sie es mit weiteren Schätzwerten, wenn das Ergebnis kleiner als 0 oder größer als 1 ist.

Wenn @EFFBELA das Ergebnis nach 30 Berechnungsdurchgängen nicht mit einer Genauigkeit von 0,0000001 berechnen kann, ist das Ergebnis FEHLER. Wenn Ihre Schätzungen weiter FEHLER ergeben, benutzen Sie @NETAKTWERT, um einen besseren Schätzwert zu ermitteln. Wenn @NETAKTWERT einen positiven Wert ergibt, ist Ihre Schätzung zu niedrig. Wenn @NETAKTWERT einen negativen Wert ergibt, ist Ihre Schätzung zu hoch. @NETAKTWERT ergibt 0, wenn Ihre Schätzung richtig ist.

Der Zeitraum, mit dem der Schätzwert berechnet wird, muß identisch mit dem Zeitraum für Laufzeit sein; wenn Sie z.B. eine monatliche Zahlung berechnen, geben Sie den Zinssatz und die Laufzeit in Monaten an. Im allgemeinen bedeutet dies, daß Sie den Zinssatz durch 12 dividieren müssen und die Anzahl von Jahren in Laufzeit mit 12 multiplizieren müssen.

Beispiele

Sie haben 6.000 DM auf ein Konto einbezahlt und möchten acht Jahre lang monatlich DM 100 abheben. Um den Zinssatz zu berechnen, der erforderlich ist, um diesen Betrag abnehmen zu können, benutzen Sie folgende @Funktion:

@EFFBELA(96;100;6000;0;0;0,01) = 0,010623 oder 1,06% monatlich bezahlt.

Ähnliche @Funktionen

@NETAKTWERT berechnet den aktuellen Nettowert einer Reihe von zukünftigen Cash-flows. @AKTWERT und @AWERT berechnen den aktuellen Wert einer Rente auf der Grundlage einer Reihe gleicher Zahlungen. @ZUKWERT und @ZWERT berechnen den zukünftigen Wert einer Rente. @ZINS berechnet den periodischen Zinssatz, der erforderlich ist, damit eine Investition auf einen zukünftigen Wert anwächst.

@INTZINS

@INTZINS(*Schätzwert*; *Bereich*) berechnet die interne Ertragsrate für eine Reihe von Cash-flow-Werten, die sich aus einer Investition ergeben. Die interne Ertragsrate ist die prozentuale Rate, die dem aktuellen Wert einer erwarteten künftigen Folge von Cash-flows für die ursprüngliche Investition entspricht.

Argumente

Schätzwert ist ein Wert in Dezimalzahlen oder Prozent, der Ihrer Schätzung der internen Ertragsrate entspricht. In den meisten Fällen muß *Schätzwert* einer Prozentzahl zwischen 0 (0%) und 1 (100%) entsprechen. Je größer die Cash-flows, desto genauer sollte Ihr *Schätzwert* sein.

Bei *Bereich* handelt es sich um die Adresse oder den Namen eines Bereichs, in dem die Cash-flows stehen. 1-2-3 betrachtet negative Zahlen als Barabgang und positive Zahlen als Bareingang. Normalerweise stellt der erste Cash-flow-Betrag in dem Bereich eine negative Zahl (einen Barabgang) dar, der der Investition entspricht. 1-2-3 weist allen leeren Zellen und Labels in *Bereich* den Wert 0 zu und nimmt sie in der Berechnung auf.

Verwendung

Mit **@INTZINS** bestimmen Sie die Rentabilität einer Investition. Kombinieren Sie **@INTZINS** mit anderen Finanz-@Funktionen, wie z.B. **@NETAKTWERT**, um eine Investition zu kalkulieren.

Hinweise

1-2-3 geht davon aus, daß die Cash-flows in regelmäßigen, gleichen Abständen eingehen.

@INTZINS berechnet die interne Ertragsrate anhand einer Reihe von Näherungswerten, wobei *Schätzwert* den Ausgangspunkt bildet. Beginnen Sie mit einer Ihrer Meinungen nach angemessenen *Schätzwert* für die interne Ertragsrate. Da mehr als eine Lösung möglich ist, versuchen Sie es mit einem weiteren *Schätzwert*, wenn das Ergebnis kleiner als 0 oder größer als 1 ist.

Kann **@INTZINS** das Ergebnis nach 30 Berechnungsdurchgängen nicht mit einer Genauigkeit von 0,0000001 schätzen, ist das Ergebnis **FEHLER**. Wenn Ihre Schätzwerte immer wieder zu dem Ergebnis **FEHLER** führen, ermitteln Sie mit **@NETAKTWERT** einen genaueren Schätzwert. Wenn **@NETAKTWERT** einen positiven Wert zurückgibt, ist Ihre Schätzung zu niedrig. Gibt **@NETAKTWERT** einen negativen Wert zurück, so ist Ihr Schätzwert zu hoch. Wenn Ihr Schätzwert richtig ist, gibt **@NETAKTWERT** 0 zurück.

Mit **@MITTELWERT** ermitteln Sie die interne Ertragsrate, wenn Sie mehrere Raten berechnen müssen.

Beispiele

In einer Aufstellung wird die interne Ertragsrate einer ursprünglichen Investition von 10.000,00 DM gefolgt von 12 monatlichen Cash-flows von je 1.500 DM berechnet. *Schätzwert* (12,00%) steht in der gleichnamigen Zelle, und die Cash-flows werden in einem Bereich namens CASH-FLOWS dargestellt.

@INTZINS(*Schätzwert*;CASH-FLOWS)

ergibt 10,45%, die interne Ertragsrate.

Ähnliche @Funktionen

@NETAKTWERT berechnet den aktuellen Nettowert einer Reihe von zukünftigen Cash-flows. **@AKTWERT** und **@AWERT** berechnen den aktuellen Wert einer Rente auf der Grundlage von gleichen Investitionen. **@ZUKWERT** und **@ZWERT** berechnen den zukünftigen Wert einer Rente. **@ZINS** gibt den periodischen Zinssatz für eine Investition zurück, die auf einen künftigen Wert anwachsen soll.

@MINTZINS berechnet die geänderte interne Ertragsrate.

@XINTZINS berechnet die interne Ertragsrate einer Reihe von Einnahmen und Ausgaben.

@ISTDEFZUS, @ISTGELZUS, @ISTMAKRO

@ISTDEFZUS(*Name*) testet *Name* auf eine definierte globale LotusScript Funktion. Wenn *Name* eine definierte Skriptfunktion ist, ergibt **@ISTDEFZUS** den Wert 1 (wahr); Wenn *Name* keine definierte Skriptfunktion ist, ergibt **@ISTDEFZUS** den Wert 0 (falsch).

@ISTGELZUS(*Name*) testet *Name* auf einen Zusatz, der gerade in den Speicher geladen ist. Wenn *Name* ein Zusatz ist, der gerade in den Speicher geladen ist, ergibt **@ISTGELZUS** den Wert 1 (wahr); ist *Name* kein Zusatz, der gerade in den Speicher geladen ist, so ergibt **@ISTGELZUS** den Wert 0 (falsch).

@ISTMAKRO(*Name*) testet *Name* auf eine definierte globale LotusScript Subroutine. Wenn *Name* eine definierte Subroutine ist, ergibt **@ISTMAKRO** den Wert 1 (wahr); ist *Name* keine definierte Subroutine, so ergibt **@ISTMAKRO** den Wert 0 (falsch).

Argumente

Name ist der Name der zu testenden Zusatz-Arbeitsmappendatei (ohne die Erweiterung .12A) oder der zu testenden Funktion oder Subroutine, in Form einer Zeichenfolge eingegeben.

Hinweise

@ISTGELZUS ergibt nur 1 (wahr) für Zusatzanwendungen, die Sie mit Datei - Zusätze - Zusätze verwalten geladen haben. **@ISAPP** gibt für jeden geladenen Zusatz 1 zurück, selbst wenn dieser nur **@**Funktionen enthält.

@ISAAF und **@ISMACRO** geben 1 (wahr) zurück, wenn Sie eine Funktion lediglich mit ihrem Namen angeben und sich die Funktion in der aktuellen Arbeitsmappe oder einem geladenen Zusatz befindet. Um eine andere Arbeitsmappe auf eine Funktion zu testen, geben Sie den Datei- und den Funktionsnamen an, z.B.

"<<Datei1>>Funktion1".

1-2-3 erkennt Funktionen und Subroutinen nur, wenn ihre Argumente und Return-Typen legal sind. Bei Fließkommazahlen erkennt 1-2-3 den Typ "Single-" nicht. Alle Funktionen und Subroutinen müssen den Typ "Double" verwenden.

Bereiche sind gültige Argumente. Sie müssen allerdings als Varianzargumente angegeben werden. 1-2-3 Objektnamen sind keine gültigen Argumenttypen.

Beispiele

@ISTDEFZUS("Grad") = 1, wenn **@GRAD** eine definierte Zusatz-Skriptfunktion ist.

@ISTDEFZUS("dsumme") = 0, da **@DSUMME** eine 1-2-3 **@**Funktion und keine Zusatz-Skriptfunktion ist.

@ISTGELZUS("Finanz") = 1, wenn ein Zusatz namens FINANZ gerade in den Speicher geladen ist.

@ISTMAKRO("Gehälter") = 1, wenn {GEHÄLTER} eine definierte Zusatz-Skript-Subroutine ist.

@ISTFEHLER

@ISTFEHLER(x) testet x auf den Wert FEHLER. Wenn x FEHLER ist, ergibt @ISTFEHLER den Wert 1 (wahr); ist x nicht FEHLER, so ergibt @ISTFEHLER den Wert 0 (falsch).

Argumente

x ist ein beliebiger Wert, Ort, eine Zeichenfolge oder eine Bedingung.

Hinweise

Mit @ISTFEHLER blockieren Sie Fehler, die sich aus einer Division durch null ergeben. So testet z.B. die Formel @WENN(@ISTFEHLER(A1/A2);0;A1/A2) das Ergebnis der Division A1/A2 (der Inhalt der Zelle A1 dividiert durch den Inhalt der Zelle A2). Wenn das Ergebnis der Wert FEHLER ist, ergibt die Formel 0. Ergibt sich ein beliebiger anderer Wert, so gibt die Formel dieses Ergebnis zurück.

Beispiele

Das Unterprogramm TEST_MENGE besteht aus drei kurzen Unterprogrammen, die die Einträge in den Zellen MENGE und PREIS überprüfen. TEST_MENGE überprüft den Eintrag in MENGE auf einen Wert. Wenn es sich um einen Wert handelt, springt der Makro zu dem Unterprogramm TEST_FEHLERNV. Enthält MENGE keinen Wert, so fordert NEU_MENGE einen neuen Eintrag an und geht zu TEST_MENGE.

TEST_FEHLERNV benutzt @ISTFEHLER, um festzustellen, ob MENGE den Wert FEHLER enthält. Gibt @ISTFEHLER 1 (wahr) zurück, wird ein neuer Wert angefordert. Enthält MENGE nicht den Wert FEHLER und enthält PREIS nicht den Wert NV, so multipliziert das Unterprogramm die Werte in den beiden Zellen und gibt das Ergebnis in der Zelle GESAMT zurück.

```
...
TEST_MENGE      {WENN @ISTZAHL(MENGE)}{SPRUNG TEST_FEHLERNV}
NEU_MENGE       {ZAHLENEINTRAG "Bitte Menge eingeben: ";MENGE}
                {SPRUNG TEST_MENGE}
TEST_FEHLERNV   {WENN @ISTFEHLER(MENGE)}{SPRUNG NEU_MENGE}
                {WENN @ISTNV(PREIS)}{ZAHLENEINTRAG "Neuen Preis eingeben: ";PREIS}{SPRUNG
TEST_MENGE}
                {GEHEZU}GESAMT~+MENGE*Preis~
...
```

Ähnliche @Funktionen

@ISTNV testet auf den Wert NV.

@ISTNV

@ISTNV(x) testet x auf den Wert NV. Wenn x der Wert NV ist, ergibt @ISTNV 1 (wahr); ist x nicht NV, so ergibt @ISTNV den Wert 0 (falsch).

Argumente

x ist ein beliebiger Wert, Ort, eine Zeichenfolge oder eine Bedingung.

Beispiele

Das Unterprogramm TEST_MENGE besteht aus drei kurzen Unterprogrammen, die die Einträge in den Zellen MENGE und PREIS überprüfen. TEST_MENGE überprüft den Eintrag in MENGE auf einen Wert; wenn es sich um einen Wert handelt, springt der Makro zu dem Unterprogramm TEST_FEHLERNV. Enthält MENGE keinen Wert, so fordert NEU_MENGE einen neuen Eintrag an und geht dann zu TEST_MENGE.

TEST_FEHLERNV benutzt @ISTNV, um festzustellen, ob PREIS den Wert FEHLER enthält. Gibt @ISTNV den Wert 1 (wahr) zurück, so wird ein neuer Wert angefordert. Enthält PREIS den Wert NV nicht und enthält MENGE nicht den Wert FEHLER, so multipliziert das Unterprogramm die Werte in den beiden Zellen und gibt das Ergebnis in die Zelle GESAMT ein.

```
...
TEST_MENGE      {WENN @ISTZAHL(MENGE)}{SPRUNG TEST_FEHLERNV}
NEU_MENGE       {ZAHLENEINTRAG "Bitte Menge eingeben: ";MENGE}
                 {SPRUNG TEST_MENGE}
TEST_FEHLERNV   {WENN @ISTFEHLER(MENGE)}{SPRUNG NEU_MENGE}
                 {WENN @ISTNV(PREIS)}{ZAHLENEINTRAG "Neuen Preis eingeben: ";PREIS}{SPRUNG
TEST_MENGE}
                 {GEHEZU}GESAMT~+MENGE*PREIS~
...

```

Ähnliche @Funktionen

@ISTFEHLER testet auf den Wert FEHLER.

@ISTZAHL

@ISTZAHL(x) testet x auf einen Wert. Wenn x ein Wert, NV, FEHLER oder eine leere Zelle ist, ergibt **@ISTZAHL** den Wert 1 (wahr); ist x eine Zeichenfolge oder eine Zelle, die ein Label oder eine Formel enthält, die ein Label ergibt, so gibt **@ISTZAHL** 0 (falsch) zurück.

Argumente

x ist ein beliebiger Wert, Ort, eine Zeichenfolge oder eine Bedingung. Wenn x ein mehrzelliger Bereich ist, ergibt **@ISTZAHL** 0 (falsch), selbst wenn die erste Zelle des Bereichs einen Wert enthält.

Beispiele

Das Unterprogramm **TEST_MENGE** besteht aus drei kurzen Unterprogrammen, die die Einträge in den Zellen **MENGE** und **PREIS** überprüfen.

TEST_MENGE benutzt **@ISTZAHL**, um zu prüfen, ob der Eintrag in **MENGE** ein Wert ist. Wenn ja, wird die Verarbeitung an das Unterprogramm **TEST_FEHLERNV** übertragen. Enthält **MENGE** keinen Wert, so fordert **NEU_MENGE** einen neuen Eintrag an.

```
...
TEST_MENGE      {WENN @ISTZAHL(MENGE)}{SPRUNG TEST_FEHLERNV}
NEU_MENGE       {ZAHLENEINTRAG "Bitte Menge eingeben: ";MENGE}
                {SPRUNG TEST_MENGE}
TEST_FEHLERNV   {WENN @ISTFEHLER(MENGE)}{SPRUNG NEU_MENGE}
                {WENN @ISTNV(PREIS)}{ZAHLENEINTRAG "Neuen Preis eingeben: ";PREIS}{SPRUNG
TEST_MENGE}
                {GEHEZU}GESAMT~+MENGE*Preis~
...

```

Ähnliche @Funktionen

@ISTFOLGE testet auf ein Label. **@ZELLE** und **@ZELLZEIGER** können ebenfalls bestimmen, ob eine Zelle einen Wert oder ein Label enthält.

@ISTBEREICH

@ISTBEREICH(x) testet x auf einen definierten Bereichsnamen oder eine gültige Bereichsadresse (eine Bereichsadresse mit Arbeitsblatt- und Spaltenbuchstaben von A bis IV und Zeilennummern von 1 bis 8192). Wenn x einen definierten Bereichsnamen oder eine gültige Bereichsadresse darstellt, ergibt @ISTBEREICH 1 (wahr); ist x kein definierter Bereichsname bzw. keine gültige Bereichsadresse, so ergibt @ISTBEREICH den Wert 0 (falsch).

Argumente

x ist eine beliebige Zeichenfolge bzw. eine Bereichsadresse.

Hinweise

@ISTBEREICH wird besonders mit @WENN benutzt, um festzustellen, ob es sich bei einem Eintrag um einen gültigen Bereichsnamen für Unterprogrammaufrufe und Sprünge mit {HOLADRESSE} handelt.

Sie können @ISTBEREICH nur mit Arbeitsmappen im Speicher benutzen.

Beispiele

@ISTBEREICH(A1) = 1 (wahr).

@ISTBEREICH(+A1) = 0 (falsch).

@ISTBEREICH(A1..C3) = 1 (wahr).

@ISTBEREICH(UMSATZ) = 1 (wahr), wenn UMSATZ ein definierter Bereichsname ist.

@ISTBEREICH(PREIS) = 0 (falsch), wenn PREIS ein undefinierter Bereichsname ist.

@ISTBEREICH(3) = 0 (falsch).

@ISTBEREICH("PROVISION") = 0 (falsch), da der Bereichsname in " " (Anführungszeichen) steht.

@ISTFOLGE

@ISTFOLGE(x) testet, ob x Text oder ein Label ist. Wenn x Text oder eine Zelle ist, die ein Label oder eine Formel enthält, die ein Label ergibt, gibt @ISTFOLGE den Wert 1 (wahr) zurück; ist x ein Wert, FEHLER, NV oder eine leere Zelle, so ergibt @ISTFOLGE den Wert 0 (falsch).

Argumente

x ist ein beliebiger Wert, Ort, eine Zeichenfolge oder Bedingung.

Beispiele

Das Unterprogramm TEST_TEXT überprüft den Inhalt der Zelle KUNDE. Enthält KUNDE ein Label (@ISTFOLGE(KUNDE) = 1), so springt das Unterprogramm zu einem neuen Unterprogramm namens BESTELLEN. Enthält KUNDE kein Label, so fordert das Unterprogramm einen neuen Eintrag an.

```
...
TEST_TEXT  {WENN @ISTFOLGE(KUNDE)}{SPRUNG BESTELLEN}
           {LABELEINTRAG "Bitte Kunden eingeben: ";KUNDE}
           {TEST_TEXT}
...
```

Ähnliche @Funktionen

@ISTZAHL testet auf einen Wert. @ZELLE und @ZELLZEIGER können ebenfalls feststellen, ob eine Zelle einen Wert oder ein Label enthält.

@KURTOSIS

@KURTOSIS(*Argument-Liste*;[*Typ*]) berechnet die Kurtosis (Wölbung) der Werte in *Argument-Liste*.

Argumente

Argument-Liste ist der Name oder die Adresse eines Bereichs, der Werte enthält. Wenn *Argument-Liste* weniger als vier Werte enthält, ergibt **@KURTOSIS** den Wert **FEHLER**.

Typ ist ein wahlweises Argument, mit dem angegeben wird, ob eine Grundgesamtheit oder eine Stichprobe berechnet wird.

| Typ | 1-2-3 berechnet |
|------------|---|
| 0 | Wölbung einer Grundgesamtheit; Vorgabe, wenn Sie das Argument weglassen |
| 1 | Wölbung einer Stichprobe |

Hinweise

Kurtosis mißt die relative Wölbung oder Flachheit einer Verteilung, verglichen mit der Normalverteilung. Eine positive Kurtosis gibt eine relativ spitze Verteilung an, während eine negative Kurtosis eine relativ flache Verteilung angibt.

Beispiele

Der DATEN-Bereich enthält folgende Werte: 2, 5, 5, 9, 1, 2, 4.

@KURTOSIS(DATEN;1) = 1,021488

@KURTOSIS(DATEN) = -0,32438

Ähnliche @Funktionen

@SCHIEFE berechnet die Schiefe der Werte in einem Bereich.

@NHÖCHST

@NHÖCHST(*Bereich*; *n*) ermittelt den *n*-höchsten Wert in *Bereich*.

Argumente

Bereich ist der Name oder die Adresse eines Bereichs, der Werte enthält.

n ist eine positive Ganzzahl. Wenn *n* größer ist als die Anzahl von Werten in *Bereich*, ergibt **@NHÖCHST** den Wert FEHLER.

Beispiele

Ein Bereich namens ERGEBNISSE enthält die folgenden Testergebnisse: 87, 85, 90, 80, 82, 92, 79, 85, 95, 86.

@NHÖCHST(ERGEBNISSE;3) ergibt 90, das dritthöchste Ergebnis.

Ähnliche @Funktionen

@NNIEDRIGST ermittelt den *n*-niedrigsten Wert in Bereich. **@MAX** und **@PURMAX** ermitteln den größten Wert in einem Bereich. **@MIN** und **@PURMIN** ermitteln den kleinsten Wert in einem Bereich.

@LINKS

@LINKS(*Zeichenfolge*;n) gibt die ersten *n* Zeichen in *Zeichenfolge* zurück.

Argumente

Zeichenfolge ist Text in " " (Anführungszeichen), eine Formel, die Text ergibt, bzw. die Adresse oder der Name einer Zelle, die ein Label oder eine Formel enthält, die ein Label ergibt.

n kann eine positive Ganzzahl oder 0 sein. Wenn *n* gleich 0 ist, ist das Ergebnis eine leere Zeichenfolge. Ist *n* größer als die Länge von *Zeichenfolge*, so gibt **@LINKS** die ganze Zeichenfolge zurück.

Hinweise

@LINKS zählt Satzzeichen und Leerstellen als Zeichen.

Mit **@LINKS** können Sie ausgewählte Teile eines Labels ab dem Anfang des Labels in eine andere Zelle kopieren (um z.B. Titel wie Dr. oder Prof. vom Namen zu trennen).

In einem Makro können Sie mit **@LINKS** Teile von Labels extrahieren, die der Benutzer eingibt, um sie in einer Datenbank zu speichern, damit Unterprogramme aufgerufen werden können oder um den Makro selbst zu ändern.

Benutzen Sie **@LINKS** mit **@FINDEN**, wenn Sie den genauen Wert von *n* nicht kennen oder wenn *n* möglicherweise variabel ist.

Beispiele

@LINKS(TELEFON;3) = die Vorwahl für die Telefonnummer in der Zelle TELEFON.

@LINKS(A1;**@FINDEN**("*";A1;0)) = der erste Name in Zelle A1 (z.B. Richard, wenn Zelle A1 den Namen Richard Schmidt enthält). Das * (Sternchen) stellt eine Leerstelle dar.

Ähnliche @Funktionen

@MITTE gibt Zeichen aus der Mitte einer *Zeichenfolge* zurück. **@RECHTS** gibt die *n* letzten Zeichen in *Zeichenfolge* zurück.

@LÄNGE

@LÄNGE(*Zeichenfolge*) zählt die Zeichen in *Zeichenfolge*.

Argumente

Zeichenfolge ist Text in " " (Anführungszeichen), eine Formel, die Text ergibt, bzw. die Adresse oder der Name einer Zelle, die ein Label oder eine Formel enthält, die ein Label ergibt.

Hinweise

@LÄNGE zählt Satzzeichen und Leerstellen als Zeichen.

Benutzen Sie @LÄNGE mit @KOMPR, um die Länge von *Zeichenfolge* ohne führende, abschließende oder aufeinanderfolgende Leerstellen zu bestimmen.

@LÄNGE wird auch in einer Anwendung benutzt, bei der Labels eine bestimmte Länge haben müssen, wie z.B. Postleitzahlen oder Bestellnummern.

Beispiele

@LÄNGE("Kühlschrank") = 11.

@LÄNGE(A5&G12) = die Gesamtsumme der Zeichen in den Zellen A5 und G12.

@LÄNGE(@KOMPR(" Herr Müller")) = 11.

@LN

@LN(x) berechnet den natürlichen Logarithmus (zur Basis e) von x.

Argumente

x ist ein Wert größer als 0.

Hinweise

Ein natürlicher Logarithmus benutzt die Zahl e (ca. 2,718282) als Basis.

Beispiele

@LN(2) = 0,693147.

@LN(@EXP(1)) = 1, da @EXP(1) = 2,718282.

Ähnliche @Funktionen

@EXP ist die Umkehrfunktion von @LN. @LOG berechnet den gewöhnlichen Logarithmus von x (zur Basis 10).

@LOG

@LOG(x) berechnet den gewöhnlichen Logarithmus von x (zur Basis 10).

Argumente

x ist ein Wert größer als 0.

Beispiele

$10^{(@LOG(8)/3)}$ = 2, die Kubikwurzel von 8.

@LOG(B3) = 0,60206, wenn Zelle B3 den Wert 4 enthält.

Ähnliche @Funktionen

@LN berechnet den natürlichen Logarithmus eines Wertes (zur Basis e).

@KLEIN

@KLEIN(*Zeichenfolge*) wandelt alle Buchstaben in *Zeichenfolge* in Kleinbuchstaben um.

Argumente

Zeichenfolge ist Text in " " (Anführungszeichen), eine Formel, die Text ergibt, bzw. die Adresse oder der Name einer Zelle, die ein Label oder eine Formel enthält, die ein Label ergibt.

Hinweise

Wenn Sie eine ASCII-Sortierfolge verwenden (Datei - Vorgaben für 1-2-3 -, unter Landesspezifische Sortierung im Register Standard), wirkt sich die Groß- und Kleinschreibung auf die Sortierfolge von Labels aus; zwei ansonsten identische Labels, die sich nur in der Schreibweise des Anfangsbuchstabens unterscheiden, können unter Umständen nicht unmittelbar hintereinander in der Liste stehen.

Beispiele

@KLEIN("Umsatz-Prognose") = umsatz-prognose.

Ähnliche @Funktionen

@GROSS wandelt alle Buchstaben in *Zeichenfolge* in Großbuchstaben um. @EIGENNAME gibt alle Wörter in *Zeichenfolge* mit großem Anfangsbuchstaben an.

@MAX

@PURMAX

@MAX(*Argument-Liste*) ermittelt den größten Wert in *Argument-Liste*.

@PURMAX(*Argument-Liste*) ermittelt den größten Wert in *Argument-Liste*, wobei Zellen mit Labels ignoriert werden.

Argumente

Argument-Liste kann eine der folgenden Angaben in einer beliebigen Kombination enthalten: Zahlen, numerische Formeln und Adressen oder Namen von Bereichen, die Zahlen oder numerische Formeln enthalten. Trennen Sie die einzelnen Elemente in *Argument-Liste* mit Argument-Trennzeichen.

Siehe auch Argumente für statistische @Funktionen.

Beispiele

Ein Bereich namens TEST enthält die folgenden Daten: -5, -7, -9, -11, Januar.

@MAX(TEST) ergibt 0, den Wert des Labels Januar, als größten Wert in TEST.

@PURMAX(TEST) ignoriert das Label Januar und gibt -5 als größten Wert in TEST zurück.

Ähnliche @Funktionen

@MIN und @PURMIN ermitteln den kleinsten Wert in einer Argument-Liste. @DMAX ermittelt den größten Wert in dem Feld einer Datenbanktabelle, das den angegebenen Kriterien entspricht. @NHÖCHST ermittelt den *n*-höchsten Wert in einer Argument-Liste.

@MEDIAN

@MEDIAN(*Argument-Liste*) ermittelt den Medianwert in *Argument-Liste*.

Argumente

Argument-Liste kann eine der folgenden Angaben in einer beliebigen Kombination enthalten: Zahlen, numerische Formeln und Adressen oder Namen von Bereichen, die Zahlen oder numerische Formeln enthalten. Trennen Sie die Elemente in *Argument-Liste* mit Argument-Trennzeichen.

Siehe auch Argumente für statistische @Funktionen.

Hinweise

Wenn *Argument-Liste* eine ungerade Anzahl von Werten enthält, berechnet @MEDIAN den mittleren Wert. Wenn *Argument-Liste* eine gerade Anzahl von Werten enthält, berechnet @MEDIAN das arithmetische Mittel der beiden mittleren Werte.

Beispiele

@MEDIAN(5;12;65;82;9) = 12

@MEDIAN(5;12;65;82;9;78) = 38,50

Ähnliche @Funktionen

@PURMEDIAN gibt den Medianwert in *Argumentliste* zurück und ignoriert dabei leere Zellen, Label und Formeln, die Label zurückgeben

@GEOMITTEL berechnet das geometrische Mittel der Werte in einer Argument-Liste. @HARMITTEL berechnet das harmonische Mittel der Werte in einer Argument-Liste. @MITTELWERT und @PURMITTELWERT berechnen den Mittelwert der Werte in einer Argument-Liste.

@MITTE

@MITTE(*Zeichenfolge*; *Startnummer*; *n*) kopiert *n* Zeichen aus *Zeichenfolge* ab dem Zeichen bei *Startnummer*.

Argumente

Zeichenfolge ist Text in " " (Anführungszeichen), eine Formel, die Text ergibt, oder die Adresse bzw. der Name einer Zelle, die ein Label oder eine Formel enthält, die ein Label ergibt.

Startnummer ist ein Zählwert. Wenn *Startnummer* größer als die Länge von *Zeichenfolge* ist, ist das Ergebnis von @MITTE eine leere Zeichenfolge.

n ist eine beliebige positive Ganzzahl oder 0. Ist *n* gleich 0, so ist das Ergebnis von @MITTE eine leere Zeichenfolge. Ist *n* größer als die Länge von *Zeichenfolge*, so gibt 1-2-3 alle Zeichen von *Startnummer* bis zum Ende der *Zeichenfolge* zurück.

Hinweise

@MITTE zählt Satzzeichen und Leerstellen als Zeichen.

Benutzen Sie eine große Zahl für *n*, wenn Sie die Länge von *Zeichenfolge* nicht kennen; 1-2-3 ignoriert die zusätzlichen Leerzeichen und gibt den ganzen Text ab *Startnummer* zurück.

Um einen Teil eines Labels auszugeben, wenn Sie dessen *Startnummer* nicht kennen, benutzen Sie @MITTE mit @FINDEN.

@MITTE wird besonders in Makros benutzt, um Teile von Labels zu speichern, die der Benutzer eingibt, um Unterprogrammaufrufe zu generieren oder um den Makro selbst zu ändern.

Beispiele

@MITTE("Tages-Konto-Stand";6;7) = Konto.

Ähnliche @Funktionen

@LINKS gibt die ersten *n* Zeichen von *Zeichenfolge* zurück, während @RECHTS die *n* letzten Zeichen in *Zeichenfolge* zurückgibt.

@MIN

@PURMIN

@MIN(*Argument-Liste*) ermittelt den kleinsten Wert in *Argument-Liste*.

@PURMIN(*Argument-Liste*) ermittelt den kleinsten Wert in *Argument-Liste*, wobei alle Zellen ignoriert werden, die Labels enthalten.

Argumente

Argument-Liste kann eine der folgenden Angaben in beliebiger Kombination enthalten: Zahlen, numerische Formeln und Adressen oder Namen von Bereichen, die Zahlen oder numerische Formeln enthalten. Trennen Sie die einzelnen Elemente in *Argument-Liste* durch Argument-Trennzeichen.

Siehe auch Argumente für statistische @Funktionen.

Beispiele

Ein Bereich namens TEST enthält die folgenden Einträge: 5, 7, 9, 11, Januar.

@MIN(TEST) ergibt 0, den Wert des Labels Januar, als kleinsten Wert in TEST.

@PURMIN(TEST) ignoriert das Label Januar und gibt 5 als kleinsten Wert in TEST zurück.

Ähnliche @Funktionen

@MAX und **@PURMAX** ermitteln den größten Wert in *Argument-Liste*. **@DMIN** ermittelt den kleinsten Wert in dem Feld einer Datenbank, das den von Ihnen angegebenen Kriterien entspricht.

@MINUTE

@MINUTE(*Zeitseriennummer*) gibt die Minuten, d.h. einen Wert von 0 bis 59, aus *Zeitseriennummer* zurück.

Argumente

Zeitseriennummer ist ein Wert von 0,000000 (Mitternacht) bis 0,999988 (23:59:59). Im allgemeinen wird *Zeitseriennummer* von einer anderen Zeit-@Funktion geliefert.

Hinweise

Der Minutenanteil wird besonders in Berechnungen benutzt, bei denen nur Minuten verwendet werden, wie z.B. die Zeit, die seit dem Starten einer Anwendung vergangen ist.

Beispiele

@MINUTE(0,333) = 59, da 0,333 die *Zeitseriennummer* für 7:59:31 ist.

@MINUTE(**@ZEIT**(11;15;45)) = 15, da 15 das *Minuten*-Argument bei **@ZEIT**(11;15;45) ist.

Ähnliche @Funktionen

@STUNDE gibt die Stunde und **@SEKUNDE** die Sekunden aus einer *Zeitseriennummer* zurück.

@MINTZINS

@MINTZINS(*Bereich*; *Finanz_Satz*; *Reinvest_Satz*; [*Typ*]) berechnet die geänderte interne Ertragsrate (den Gewinn) für eine Reihe von Cash-flow-Werten, die sich aus einer Investition ergeben.

Die interne Ertragsrate ist der Prozentsatz, der dem aktuellen Wert einer erwarteten zukünftigen Reihe von Cash-flows für die ursprüngliche Investition entspricht.

Argumente

Bereich ist die Adresse oder der Name eines Bereichs, der die Cash-flows enthält. 1-2-3 betrachtet negative Zahlen als Barabgang und positive Zahlen als Bareingang. *Bereich* muß mindestens einen positiven und einen negativen Wert enthalten.

Normalerweise ist der erste Cash-flow-Betrag in *Bereich* eine negative Zahl (ein Barabgang), der der Investition entspricht. 1-2-3 weist allen leeren Zellen und Labels in *Bereich* den Wert 0 zu und nimmt diese in der Berechnung auf.

Finanz_Satz ist der Zinssatz, der auf die Gelder in Cash-flows bezahlt wird.

Reinvest_Satz ist der Zinssatz, den Sie für die Cash-flows erhalten, wenn Sie diese wieder investieren.

Finanz_Satz und *Reinvest_Satz* können beliebige Werte darstellen.

Typ bestimmt den Zeitpunkt der Cash-flows. *Typ* ist eine Ganzzahl der aus folgenden Tabelle:

| Typ | Zeitpunkt der Cash-flows |
|------------|---|
| 0 | Am Anfang jeder Periode; Vorgabe, wenn Sie das Argument weglassen |
| 1 | Am Ende jeder Periode |

Hinweise

Mit @MINTZINS bestimmen Sie die Rentabilität einer Investition. Kombinieren Sie @MINTZINS mit anderen Finanz-@Funktionen, wie z.B. @NETAKTWERT, um eine Investition zu bewerten.

1-2-3 geht davon aus, daß die Cash-flows in regelmäßigen gleichen Abständen eingehen.

Das wahlweise Argument *Typ* für @MINTZINS wird nur in Version 5 und später unterstützt. Wenn Sie eine @MINTZINS Formel speichern, die ein *Typ*-Argument enthält und die Arbeitsmappe dann in einer früheren Version öffnen, ergibt die Formel FEHLER

Beispiele

Ihnen gehört ein Wohnhaus. Vor fünf und sechs Jahren haben Sie 100.000 DM zu einem jährlichen Zinssatz von 9,5% aufgenommen, um das Haus zu kaufen. Die folgende Liste, die in dem Bereich ERTRAG gespeichert ist, enthält die ursprüngliche Investition und die nachfolgenden Einkünfte aus Vermietung:

-100.000

-100.000

DM 45.500

DM 47.000

DM 48.500

DM 50.000

DM 50.000

Während dieser Jahre haben Sie für die re-investierten Gewinne 11% jährlich erhalten.

@MINTZINS(ERTRAG;0,095;0,11) = 7,69%

Der Ertrag ist etwas niedriger, wenn Sie die Investition am Anfang des Jahres vornehmen.

@MINTZINS(ERTRAG;0,095;0,11;1) = 7,70%

Ähnliche @Funktionen

@INTZINS berechnet die interne Ertragsrate. @XINTZINS berechnet die interne Ertragsrate einer Reihe von Einnahmen und Ausgaben.

@MOD

@MODULO

@MOD(x;y) und @MODULO (x;y) berechnen den Rest (Modulus) von x/y.

Argumente

x ist ein Wert. Wenn x 0 ist, ergeben @MOD und @MODULO den Wert 0.

y ist ein anderer Wert als 0.

Hinweise

Das Ergebnis von @MOD ist $x - y * @GANZZAHL(x/y)$. Das Vorzeichen des Ergebnisses (+ oder -) ist immer dasselbe wie das Vorzeichen von x.

Das Ergebnis von @MODULO ist $x - y * @ABRUNDEN(x/y)$. Das Vorzeichen des Ergebnisses (+ oder -) ist immer dasselbe wie das Vorzeichen von x.

Beispiele

@MOD(9;4) = 1

@MODULO(9;4) = 1

@MOD(-14;3) = -2

@MODULO(-14;3) = 1

Ähnliche @Funktionen

@QUOTIENT berechnet das Ergebnis von x/y, abgeschnitten auf eine Ganzzahl.

@MONAT

@MONAT(*Datumseriennummer*) ermittelt den Monat, d.h. einen Wert von 1 bis 12, aus *Datumseriennummer*.

Argumente

Datumseriennummer ist ein Wert von 1 (1. Januar 1900) bis 73050 (31. Dezember 2099).

Hinweise

Sie können eine der anderen Kalender-@Funktionen benutzen, um den Wert für *Datumseriennummer* zu liefern.

Beispiele

@MONAT(@DATUM(91;3;27)) = 3, da 3 das *Monat*-Argument bei @DATUM(91;3;27) ist.

@MONAT(20181) = 4, da die Datumseriennummer 20181 dem 02-Apr-55 entspricht.

@MONAT(@JETZT) = der laufende Monat.

Ähnliche @Funktionen

@TAG berechnet den Tag mit einer Datumseriennummer. @JAHR berechnet das Jahr mit einer Datumseriennummer.

Beispiel: @HVERWEIS

In einer horizontalen Verweistabelle namens SÄTZE (A2..E7) werden die Kosten für das Versenden eines Päckchens in verschiedene Städte aufgeführt.

@HVERWEIS("Frankfurt";SÄTZE;3), eingegeben in eine als Währung mit zwei Dezimalstellen formatierte Zelle, ergibt 24 DM, den Satz für das Versenden eines Paketes vom Typ 3 nach Frankfurt.

| A | ---- | A | ----- | B | ----- | C | ----- | D | ----- | E | -- |
|---|------|----------|-------|----------|-------|-----------|-------|----------|-------|---|----|
| 1 | | | ----- | ZIEL | ----- | | | | | | |
| 2 | Typ | London | | Paris | | Frankfurt | | New York | | | |
| 3 | 1 | DM 18,36 | | DM 19,33 | | DM 20,12 | | DM 9,29 | | | |
| 4 | 2 | DM 20,32 | | DM 21,66 | | DM 22,03 | | DM 11,25 | | | |
| 5 | 3 | DM 22,44 | | DM 23,88 | | DM 24,00 | | DM 13,25 | | | |
| 6 | 4 | DM 24,14 | | DM 25,26 | | DM 25,75 | | DM 16,85 | | | |
| 7 | 5 | DM 28,32 | | DM 29,00 | | DM 29,80 | | DM 19,54 | | | |

@W

@W(Bereich) gibt den Eintrag in der ersten Zelle von *Bereich* als Wert zurück. Wenn die Zelle ein Label enthält, gibt **@W** den Wert 0 zurück.

Argumente

Bereich ist eine Zelle oder Bereichsadresse bzw. ein Bereichsname.

Hinweise

@W kann mit jeder **@**Funktion oder Formel benutzt werden, wenn eine Zelle möglicherweise ein Label enthält und der Eintrag ein Wert sein muß. Mit **@W** verhindern Sie, daß die Auswertung von Formeln FEHLER ergibt.

@W wird auch in Makros zur Überprüfung von Benutzereinträgen benutzt.

Beispiele

+100+**@W**(B5..F5) = 885, wenn die Zelle B5 den Wert 785 enthält.

@W(A5)+**@W**(B5) ergibt 785, wenn A5 ein Label und B5 den Wert 785 enthält.

Ähnliche @Funktionen

@E gibt den Eintrag in der ersten Zelle eines Bereichs als Label zurück. **@ISTZAHL** kann feststellen, ob eine Zelle einen Wert enthält.

@NV

@NV gibt den Wert NV (nicht verfügbar) zurück.

Hinweise

@NV wird besonders beim Erstellen eines Arbeitsblattes benutzt, das Daten enthält, die noch nicht feststehen. Mit @NV kennzeichnen Sie die Zellen, in die Sie die Daten eingeben werden; Formeln, die sich auf diese Zellen beziehen, ergeben NV, bis Sie die richtigen Daten eingeben.

Mit @NV können Sie auch feststellen, welche Formeln von einer bestimmten Zelle abhängen.

Beispiele

@WENN(@ZELLE("Typ";B14)="I";@NV;B14) ergibt den Wert NV, wenn B14 leer ist.

Ähnliche @Funktionen

@FEHLER ergibt den Wert FEHLER. @ISTNV testet auf den Wert NV.

@JETZT

@JETZT berechnet die Zahl, die dem laufenden Datum und der laufenden Uhrzeit auf Ihrer Computeruhr entsprechen. Dies umfaßt sowohl eine Datumseriennummer (ganzzahliger Teil) als auch eine Zeitseriennummer (Dezimalteil).

Hinweise

Benutzen Sie @JETZT mit F2 (EDIT) und F9 (KALK), um Datum und Uhrzeit für Arbeitsblätter, die mit einem Zeitstempel versehen werden müssen, oder in Berechnungen der abgelaufenen Zeit aufzuzeichnen.

Formatieren Sie den Wert von @JETZT mit einem der Datum- oder Uhrzeitformate. Wenn Sie @JETZT als Datum formatieren, zeigt 1-2-3 nur den Datumsteil (ganzzahligen Teil) der Datum- und Zeitseriennummer an. Wenn Sie @JETZT als Uhrzeit formatieren, zeigt 1-2-3 nur den Zeiteil (Dezimalteil) der Datum- und Zeitseriennummer an. In beiden Fällen speichert und berechnet 1-2-3 die ganze Datum- und Zeitseriennummer.

1-2-3 berechnet @JETZT jedesmal neu, wenn Sie Ihr Arbeitsblatt neu berechnen.

Beispiele

@JETZT = 31050,5 am 3. Januar 1985, 12 Uhr mittags.

@JETZT = 33418,395 am 29. Juni 1991 um 9:28 Uhr.

Ähnliche @Funktionen

@HEUTE berechnet die Datumseriennummer, die dem laufenden Datum entspricht.

@NETAKTWERT

@NETAKTWERT(*Zinssatz*; *Cash-flow-Bereich*; [*Typ*]) berechnet den aktuellen Nettowert einer Reihe von künftigen Cash-flow-Werten (*Cash-flow-Bereich*), die mit einem festen periodischen *Zinssatz* bezahlt werden.

Argumente

Zinssatz ist ein Dezimal- oder Prozentwert größer als -1.

Cash-flow-Bereich ist ein Bereich, der die Cash-flows enthält.

Typ bestimmt den Zeitpunkt der Cash-flows. *Typ* ist eine Ganzzahl aus der folgenden Tabelle:

| <u>Typ</u> | <u>Zeitpunkt der Cash-flows</u> |
|------------|---|
| 0 | Am Anfang jeder Periode; Vorgabe, wenn Sie das Argument weglassen |
| 1 | Am Ende jeder Periode |

Hinweise

Mit @NETAKTWERT werten Sie eine Investition aus oder vergleichen verschiedene Investitionen miteinander.

@NETAKTWERT berechnet die ursprüngliche Investition, die benötigt wird, um einen bestimmten Cash-flow bei einem bestimmten Zinssatz zu erhalten.

@NETAKTWERT ergibt FEHLER, wenn der *Cash-flow-Bereich* mehr als eine Zeile oder Spalte enthält. So ergibt @NETAKTWERT z.B. FEHLER, wenn *Cash-flow-Bereich* A1..D25 entspricht, ergibt jedoch nicht FEHLER, wenn der *Bereich* A1..D1 (eine einzelne Zeile) oder A1..A25 (eine einzelne Spalte) ist.

Das wahlweise *Typ*-Argument für @NETAKTWERT wird nur in Version 5 und später unterstützt. Wenn Sie eine @NETAKTWERT Formel speichern, die ein *Typ*-Argument enthält, und die Arbeitsmappe dann in einer früheren 1-2-3 Version öffnen, ergibt die Formel FEHLER.

Beispiele

In diesem Beispiel wird @NETAKTWERT verwendet, um den aktuellen Wert einer Reihe von ungleichen, unregelmäßigen Zahlungen zu diskontieren, die bei einem jährlichen Zinssatz von 11,5% investiert werden.

Cash-flow-Bereich ist eine Liste mit Cash-flows, einer pro Monat über eine Dauer von 12 Monaten, in einem Bereich namens SUMME:

DM 0,00
DM 0,00
DM 2.500,00
DM 2.500,00
DM 3.000,00
DM 5.000,00
DM 6.000,00
DM 9.000,00
DM 3.000,00
DM 2.500,00
DM 0,00
DM 7.500,00

Damit @NETAKTWERT mit der richtigen Anzahl von Perioden rechnet, müssen auch die Monate, in denen keine Zahlung erfolgt, in diesen *Cash-flow-Bereich* aufgenommen werden. Da die Zahlungen monatlich erfolgen, ist es erforderlich, den monatlichen *Zinssatz* (Diskontsatz) bei @NETAKTWERT anzugeben:

$$0,115/12 = 0,96$$

$$\text{@NETAKTWERT(DISKONT;SUMME)} = 38.084,13 \text{ DM.}$$

Sie erhalten ein anderes Ergebnis, wenn der Cash-flow-Abgang am Anfang jeder Periode erfolgt.

$$\text{@NETAKTWERT(DISKONT;SUMME;1)} = 38.449,10 \text{ DM.}$$

Ähnliche @Funktionen

@AKTWERT berechnet den aktuellen Wert einer Annuität auf der Grundlage einer Reihe von gleichen Zahlungen.

@ZUKWERT berechnet den zukünftigen Wert einer Annuität.

@XNETAKTWERT berechnet den aktuellen Nettowert einer Reihe von Einnahmen und Ausgaben.

@UNGERADE

@UNGERADE(x) rundet den Wert x zur nächsten ungeraden Ganzzahl auf.

Argumente

x ist ein beliebiger Wert. Wenn x eine ungerade Ganzzahl ist, ergibt @UNGERADE den Wert x.

Beispiele

@UNGERADE(3,25) = 5

@UNGERADE(3) = 3

@UNGERADE(-3,25) = -5

Ähnliche @Funktionen

@GERADE rundet einen Wert zur nächsten geraden Ganzzahl auf. @RUNDEN, @ABRUNDEN und @AUFRUNDEN runden einen Wert auf eine bestimmte Anzahl von Dezimalstellen. @RUNDENM rundet einen Wert auf ein bestimmtes Vielfaches. @GANZZAHL schneidet einen Wert ab, wobei der Dezimalteil ignoriert wird. @ABKÜRZEN schneidet einen Wert auf eine bestimmte Dezimalstelle ab.

@PERCENTIL

@PERCENTIL(*x*; *Bereich*) berechnet das *x*-te Perzentil einer Stichprobe unter den Werten in *Bereich*.

Argumente

x ist das zu ermittelnde Perzentil. *x* ist ein Wert von 0 bis 1.

Bereich ist der Name oder die Adresse des Bereichs, in dem Werte stehen.

1-2-3 weist allen Labels in *Bereich* den Wert 0 zu und nimmt sie in der **@PERCENTIL** Berechnung auf. 1-2-3 ignoriert leere Zellen in *Bereich*.

Beispiele

Ein Bereich namens ERGEBNISSE enthält die folgenden Testergebnisse: 87, 85, 90, 80, 82, 92, 79, 85, 95, 86. Nun möchten Sie das Ergebnis bei dem 90-ten Perzentil ermitteln.

@PERCENTIL(0,9;ERGEBNISSE) = 92,3

Ähnliche @Funktionen

@PRANG ermittelt das Perzentil in einem Bereich, der mit einem Wert verknüpft ist.

@DEZIL gibt ein gegebenes Dezil zurück. **@QUARTIL** gibt ein gegebenes Quartil zurück.

@PERMUT

@PERMUT($n;r$) berechnet die Reihe von geordneten Folgen (Permutationen) von r Objekten, die unter einer Gesamtzahl von n Objekten gewählt werden können.

Argumente

n ist eine beliebige positive Ganzzahl oder 0.

r ist eine beliebige positive Ganzzahl oder 0. r darf nicht größer sein als n .

Wenn n und r keine Ganzzahlen sind, schneidet 1-2-3 sie auf Ganzzahlen ab.

Beispiele

Prüfungen, die für 9:00, 10:00 und 11:00 Uhr festgelegt sind, werden von drei von insgesamt fünf Prüfern überwacht. Mit der folgenden Formel wird die Anzahl von Möglichkeiten berechnet, die Prüfer zuzuweisen:

$$\text{@PERMUT}(5;3) = 60$$

Ähnliche @Funktionen

@KOMBIN berechnet die Möglichkeiten, mit denen r unter n ungeachtet der Reihenfolge gewählt werden kann.

@PI

@PI erzeugt den Wert p (berechnet mit 3,14159265358979324). Der Wert p ist die Verhältniszahl eines Kreisumfangs zum Durchmesser.

Beispiele

@PI = 3,1415926536.

@PI*4^2 = 50,26548, die Fläche eines Kreises mit dem Radius 4.

@RATE

@ANNUITÄT

@RATE(*Kapital*; *Zinssatz*; *Laufzeit*,) berechnet die Abzahlung eines Darlehens (*Kapital*) auf der Grundlage eines bestimmten periodischen *Zinssatzes* innerhalb einer bestimmten *Laufzeit*.

@ANNUITÄT(*Kapital*; *Zinssatz*; *Laufzeit*; [*Typ*]; [*Zukwert*]) berechnet die Abzahlung eines Darlehens (*Kapital*) auf der Grundlage eines bestimmten *Zinssatzes* innerhalb einer bestimmten *Laufzeit*. @ANNUITÄT berechnet entweder eine vorschüssige oder nachschüssige Rente, je nach dem für *Typ* angegebenen Wert.

Argumente

Kapital und *Laufzeit* sind Werte.

Zinssatz ist ein Dezimal- oder Prozentwert größer als -1.

Typ ist ein wahlweises Argument, mit dem angegeben wird, ob eine nachschüssige oder vorschüssige Rente berechnet wird.

| <u>Typ</u> | <u>1-2-3 berechnet</u> |
|------------|--|
| 0 | Nachschüssige Rente (Zahlungen erfolgen am Ende der Periode); Vorgabe, wenn Sie das Argument weglassen |
| 1 | Vorschüssige Rente (Zahlungen erfolgen am Anfang der Periode) |

Zukwert ist ein wahlweises Argument, mit dem der zukünftige Wert einer Reihe von Zahlungen angegeben wird. *Zukwert* kann ein beliebiger Wert sein. Wenn Sie das *Zukwert*-Argument weglassen, benutzt 1-2-3 den Wert 0.

Sie können kein wahlweises Argument ohne die davorstehenden wahlweisen Argumente benutzen.

Hinweise

Der Zeitraum, für den der *Zinssatz* berechnet wird, muß derselbe Zeitraum wie für *Laufzeit* sein. Wenn Sie z.B. eine monatliche Zahlung berechnen, geben Sie den *Zinssatz* und die *Laufzeit* in monatlichen Schritten ein. Im allgemeinen bedeutet dies, daß Sie den *Zinssatz* durch 12 dividieren und die Anzahl von Jahren in *Laufzeit* mit 12 multiplizieren müssen.

Beispiele

Sie möchten ein Darlehen in Höhe von 8.000 DM mit einer Laufzeit von 3 Jahren und einem jährlichen Zinssatz von 10,5% aufnehmen, der monatlich bezahlt wird. Die Zahlungen sind am letzten Tag jedes Monats fällig. Nun möchten Sie die monatliche Zahlung berechnen:

@RATE(8000;0,105/12;36) = 260,02 DM

Wenn Zahlungen am ersten Tag des Monats fällig sind, benutzen Sie folgende @Funktion:

@ANNUITÄT(8000;0,105/12;36;1;0) = 257,76 DM

Ähnliche @Funktionen

@RATE2 berechnet die Zahlungen für ein Darlehen bei gegebenem Zinssatz für eine festgelegte Anzahl von Perioden und bei vorschüssiger Rente. @RATEZ berechnet den Zinsanteil von periodischen Zahlungen. @SPI berechnet den Zinsanteil von periodischen Zahlungen für ein Darlehen bei gleichbleibender Tilgungssumme je Zahlungsperiode.

@ZINSBETRAG berechnet den kumulativen Zinsanteil der periodischen Zahlung für eine Investition. @TILGUNG berechnet den Kapitalteil der periodischen Zahlung für eine Investition. @ANN berechnet die Anzahl von Zahlungsperioden für eine Investition.

@POISSON

@POISSON(*x*; *Mittel*; [*Kumulativ*]) berechnet die Poisson-Verteilung.

Argumente

x ist die Anzahl von beobachteten Ereignissen und entspricht einer positiven Ganzzahl oder 0.

Mittel ist die erwartete Anzahl von Ereignissen und entspricht einer positiven Ganzzahl oder 0.

Wenn *x* und *Mittel* keine Ganzzahlen sind, schneidet 1-2-3 sie auf Ganzzahlen ab.

Kumulativ ist ein wahlweises Argument, mit dem angegeben wird, wie 1-2-3 @POISSON berechnet.

| <i>Kumulativ</i> | 1-2-3 berechnet |
|-------------------------|--|
| 0 | Die Wahrscheinlichkeit von genau <i>x</i> Ereignissen; die Vorgabe, wenn Sie das Argument weglassen |
| 1 | Die Wahrscheinlichkeit von höchstens <i>x</i> Ereignissen |

Hinweise

@POISSON berechnet die Poisson-Verteilung näherungsweise innerhalb eines Bereichs von $\pm 3 \cdot 10^{-7}$.

@POISSON wird benutzt, um die Anzahl von Ereignissen vorherzusagen, die während eines bestimmten Zeitraums auftreten, z.B. die Anzahl von Besuchern, die die Tore eines Vergnügungsparks in einer Stunde passieren.

Beispiele

Sie gehen davon aus, daß sechs Fahrzeuge in einer Stunde die Tore passieren. Um die Wahrscheinlichkeit zu berechnen, daß maximal vier Fahrzeuge in einer Stunde die Tore passieren, benutzen Sie folgende @Funktion:

@POISSON(4;6;1) = 0,285057 oder 28,51%

Um die Wahrscheinlichkeit zu berechnen, daß genau vier Fahrzeuge in einer Stunde die Tore passieren, benutzen Sie folgende @Funktion:

@POISSON(4;6) = 0,133853 oder 13,39%

@PRANG

`@PRANG(x;Bereich;[Stellen])` ermittelt das Percentil von x unter den Werten in *Bereich*.

Argumente

x ist ein beliebiger Wert.

Bereich ist der Name oder die Adresse eines Bereichs, der Werte enthält.

Stellen ist ein wahlweises Argument, das die Anzahl von Dezimalstellen angibt, auf die das Ergebnis von @PRANG gerundet werden muß. *Stellen* ist ein Wert von 0 bis 100. Wenn Sie das *Stellen*-Argument weglassen, benutzt 1-2-3 den Wert 2.

Hinweise

Wenn x kein Wert in *Bereich* ist, weist 1-2-3 die 0-te Percentil-Position dem niedrigsten Wert in *Bereich* und die 100-te Percentil-Position dem höchsten Wert in *Bereich* zu und interpoliert.

Beispiele

Ein Bereich namens ERGEBNISSE enthält die folgenden Testergebnisse: 87, 85, 90, 80, 82, 92, 79, 85, 95, 86. Nun möchten Sie das Percentil für ein Ergebnis von 90 berechnen.

`@PRANG(90;ERGEBNISSE)` = 0,78 oder 78%.

Ähnliche @Funktionen

`@PERCENTIL` berechnet ein Stichproben-Percentil für die Werte in einer Liste mit Werten.

@KURS

@KURS(*Valutierungstermin*; *Fälligkeit*; *Nominalzins*; *Rendite*; [*Rückzahlung*]; [*Zinstermine*]; [*Zinstage*]) berechnet den Kurs pro 100 DM Nennwert für Anleihen mit periodischen Zinszahlungen.

Argumente

Valutierungstermin ist das Abrechnungsdatum für die Anleihe. *Valutierungstermin* ist eine Datumseriennummer.

Fälligkeit ist das Fälligkeitsdatum einer Anleihe. *Fälligkeit* ist eine Datumseriennummer. Wenn die *Fälligkeit* kleiner als oder gleich *Valutierungstermin* ist, ergibt @KURS den Wert FEHLER.

Nominalzins ist der jährliche Nominalzins der Anleihe. *Nominalzins* ist ein beliebiger positiver Wert oder 0.

Rendite ist die jährliche Rendite. *Rendite* ist ein beliebiger positiver Wert.

Rückzahlung ist ein wahlweises Argument, mit dem der Rückzahlungswert der Anleihe pro 100 DM Nennwert angegeben wird. *Rückzahlung* ist ein beliebiger positiver Wert oder 0. Wenn Sie das *Rückzahlung*-Argument weglassen, benutzt 1-2-3 den Wert 100.

Zinstermine ist ein wahlweises Argument, mit dem die Zinstermine pro Jahr angegeben werden. *Zinstermine* ist ein Wert aus der folgenden Tabelle:

| <u>Zinstermine</u> | <u>Häufigkeit</u> |
|--------------------|--|
| 1 | Jährlich |
| 2 | Halbjährlich; Vorgabe, wenn Sie das Argument weglassen |
| 4 | Vierteljährlich |
| 12 | Monatlich |

Zinstage ist ein wahlweises Argument, mit dem angegeben wird, auf welcher Basis die Zinstage berechnet werden. *Zinstage* ist ein Wert aus der folgenden Liste:

| <u>Zinstage</u> | <u>Berechnungsbasis</u> |
|-----------------|--|
| 0 | 30/360; Vorgabe, wenn Sie das Argument weglassen |
| 1 | Anzahl/Anzahl |
| 2 | Anzahl/360 |
| 3 | Anzahl/365 |
| 4 | Europäisch 30/360 |

Sie können ein wahlweises Argument nicht ohne die davorstehenden wahlweisen Argumente benutzen.

Beispiele

Eine Anleihe hat einen Valutierungstermin vom 1. Juli 1993 und einen Fälligkeitstermin vom 1. Dezember 1998. Der halbjährliche Nominalzins beläuft sich auf 5,50% und die jährliche Rendite auf 5,61%. Die Zinstage für die Obligation werden auf einer 30/360 Tagesbasis berechnet.

Um den Kurs der Anleihe zu berechnen, benutzen Sie folgende @Funktion:

@KURS(@DATUM(93;7;1);@DATUM(98;12;1);0,055;0,0561;100;2;0) = 99,49 DM

Ähnliche @Funktionen

@STÜCKZINS berechnet den Stückzins für Anleihen, für die periodische Zinsen bezahlt werden. @RENDITE berechnet die Rendite für Anleihen, für die periodische Zinsen bezahlt werden. @LAUFZEIT berechnet die jährliche Laufzeit und @MLAUFZEIT die modifizierte jährliche Laufzeit von Anleihen, für die regelmäßig Zinsen bezahlt werden. @KURS2 berechnet den Kurs pro 100 Yen Nennwert für Wertpapiere mit periodischen Zinszahlungen auf der Grundlage japanischer Konventionen.

@EIGENNAME

@EIGENNAME(*Zeichenfolge*) schreibt den ersten Buchstaben jedes Wortes in *Zeichenfolge* groß und die restlichen Buchstaben klein.

Argumente

Zeichenfolge kann Text in " " (Anführungszeichen), eine Formel, die eine Zeichenfolge ergibt, bzw. die Adresse oder der Name einer Zelle sein, die ein Label oder eine Formel enthält, die ein Label ergibt.

Hinweise

@EIGENNAME wird benutzt, wenn Sie Daten aus verschiedenen Quellen kombinieren und die Labels in der ganzen Arbeitsmappe konsistent sein sollen. Verwenden Sie **@EIGENNAME** in einer Datenbank, damit die Schreibweise der Namen richtig ist, bevor Sie die Namen sortieren oder bevor Sie mit den Namen Adreßetiketten erstellen.

Wenn Sie eine ASCII-Sortierfolge verwenden (Datei - Vorgaben für 1-2-3 -, unter Landesspezifische Sortierung im Register Standard), wirkt sich die Groß- und Kleinschreibung auf die Sortierfolge von Labels aus; zwei ansonsten identische Labels, die sich nur in der Schreibweise des Anfangsbuchstabens unterscheiden, können unter Umständen nicht unmittelbar hintereinander in der Liste stehen.

Beispiele

@EIGENNAME(A7&" ; "&G7) ergibt Hans Müller; 7000 Stuttgart, wenn A7 das Label HANS MÜLLER und G7 das Label 7000 Stuttgart enthält. Beachten, Sie, daß das ; (Semikolon) in Anführungszeichen steht und deshalb als literale Zeichenfolge und nicht als Argument-Trennzeichen benutzt wird.

Ähnliche @Funktionen

@KLEIN setzt alle Buchstaben in *Zeichenfolge* in Kleinbuchstaben um; **@GROSS** setzt alle Buchstaben in *Zeichenfolge* in Großbuchstaben um.

@AKTWERT

@AWERT

@AKTWERT(*Zahlungen*; *Zinssatz*; *Laufzeit*) berechnet den aktuellen Wert einer Investition auf der Grundlage gleicher *Zahlungen* zu einem periodischen *Zinssatz* über eine bestimmte *Laufzeit*.

@AWERT(*Zahlungen*; *Zinssatz*; *Laufzeit*; [*Typ*]; [*Zukwert*]) berechnet den aktuellen Wert einer Investition mit einem bestimmten *Zukwert* auf der Grundlage gleicher *Investitionen* mit einem periodischen *Zinssatz* über eine bestimmte *Laufzeit*. @AWERT berechnet entweder eine vorschüssige oder nachschüssige Rente, je nach Wert, der für *Typ* angegeben wird.

Argumente

Zahlungen und *Laufzeit* sind Werte.

Zinssatz ist ein Dezimal- oder Prozentwert größer als -1.

Typ ist ein wahlweises Argument, mit dem angegeben wird, ob eine nachschüssige oder vorschüssige Rente berechnet wird.

| <u>Typ</u> | <u>1-2-3 berechnet</u> |
|------------|--|
| 0 | Nachschüssige Rente (die Zahlungen erfolgen am Ende einer Periode); Vorgabe, wenn Sie das Argument weglassen |
| 1 | Vorschüssige Rente (die Zahlungen erfolgen am Anfang einer Periode) |

Zukwert ist ein wahlweises Argument, mit dem der zukünftige Wert einer Reihe von Zahlungen angegeben wird. *Zukwert* kann ein beliebiger Wert sein. Wenn Sie das Argumente *Zukwert* weglassen, benutzt 1-2-3 den Wert 0.

Sie können ein wahlweises Argument nicht ohne die davorstehenden wahlweisen Argumente benutzen.

Hinweise

Der Zeitraum, der zur Berechnung des *Zinssatzes* benutzt wird, muß identisch mit dem Zeitraum für *Laufzeit* sein; wenn Sie z.B. eine monatliche Bezahlung berechnen, geben Sie den Zinssatz und die Laufzeit in Monaten ein. Dies bedeutet im allgemeinen, daß Sie den Zinssatz durch 12 dividieren und die Anzahl von Jahren in *Laufzeit* mit 12 multiplizieren müssen.

Benutzen Sie @AKTWERT, um eine Investition zu berechnen oder um eine Investition mit anderen Investitionen zu vergleichen. @AKTWERT wird besonders für den Vergleich unterschiedlicher Investitionen benutzt, z.B. für den Vergleich einer Investition mit einer einmaligen Zahlung aus einem Rentenfonds mit einer Reihe von periodischen Zahlungen. Wenn Sie @AKTWERT mit @RATE kombinieren, können Sie eine Amortisationstabelle erstellen.

@AKTWERT ergänzt @RATE: @AKTWERT errechnet für Sie die Höhe der maximalen Kreditaufnahme, wobei berechnet wird, welche Summe Sie monatlich aufbringen können. Im Gegensatz dazu errechnet @RATE für Sie die Höhe Ihrer monatlichen Zahlung, wobei berücksichtigt wird, wie hoch der Kredit ist, den Sie aufnehmen möchten.

Beispiele

Sie haben 1.000.000 DM geerbt. Nun können Sie entweder 20 jährliche Zahlungen in Höhe von 50.000 DM am Ende eines jeden Jahres oder eine einzige Zahlung in Höhe von 400.000 DM erhalten, anstelle der Rente in Höhe von 1.000.000 DM. Nun möchten Sie feststellen, welche Möglichkeit beim heutigen DM-Wert mehr wert ist.

Wenn Sie Jahreszahlungen in Höhe von 50.000 DM akzeptieren, gehen Sie davon aus, daß Sie den Betrag mit einem Zinssatz von 8%, der jährlich bezahlt wird, investieren.

@AKTWERT(50000;0,08;20) ergibt 490.907 DM, d.h. daß der Betrag von 1.000.000 DM, über 20 Jahre ausbezahlt, nach dem heutigen DM-Wert 490.907 DM wert ist.

Wenn Sie die Zahlungen am Anfang jedes Jahres erhalten, benutzen Sie folgende Formel:

@AWERT(50000;0,08;20;1) = 530.180 DM

Ähnliche @Funktionen

@ZUKWERT und @ZWERT berechnen den zukünftigen Wert einer Investition auf der Grundlage einer Reihe von gleichen Investitionen. @NETAKTWERT berechnet den aktuellen Nettowert einer Reihe von zukünftigen Cash-flows. @RATE und @ANNUITÄT berechnen die Bezahlung eines Darlehens zu einem bestimmten Zinssatz während einer bestimmten Anzahl von Zahlungsperioden.

@AKTWERT2(*Zahlungen*; *Zinssatz*; *Laufzeit*) berechnet den aktuellen Wert einer Investition, basierend auf einer

Reihe gleicher *Zahlungen* bei Diskontierung periodischer *Zinssätze* über die Anzahl der Perioden in *Laufzeit* bei vorschüssiger Rente. @AWERTBETRAG berechnet den aktuellen Wert eines angelegten Pauschalbetrags für eine gegebene Anzahl von Perioden bei Diskontierung eines gegebenen Zinssatzes.

@QUOTIENT

@QUOTIENT(x;y) berechnet das Ergebnis von x/y , auf eine Ganzzahl abgeschnitten.

Argumente

x ist ein Wert. Wenn x gleich 0 ist, ergibt @QUOTIENT den Wert 0.

y ist ein Wert mit Ausnahme von 0.

Beispiele

@QUOTIENT(7;3) = 2

@QUOTIENT(12,25;3,5) = 3

@QUOTIENT(-7;3) = -2

Ähnliche @Funktionen

@MOD berechnet den Rest (Modulus) von x/y .

@RADINGRAD

@RADINGRAD(*Radiant*) wandelt Radiant in Grad um.

Argumente

Radiant ist ein Wert.

Beispiele

@RADINGRAD(0,523599) = 30 Grad

Ähnliche @Funktionen

@GRADINRAD wandelt Grad in Radiant um.

@ZUFALLSZAHL

@ZUFALLSZAHL generiert eine Zufallszahl zwischen 0 und 1. 1-2-3 berechnet @ZUFALLSZAHL mit einer Genauigkeit von 17 Dezimalstellen. Sobald 1-2-3 Ihre Daten neu berechnet, generiert @ZUFALLSZAHL eine neue Zufallszahl.

Hinweise

Um den von @ZUFALLSZAHL generierten Wert in einen festen Wert umzuwandeln, drücken Sie F2 (EDIT) und danach F9 (KALK).

Um Zufallswerte in anderen numerischen Intervallen zu erzeugen, multiplizieren Sie @ZUFALLSZAHL mit der Größe des Intervalls. Benutzen Sie @RUNDEN oder @GANZZAHL mit dem Ergebnis, um ganze Zufallszahlen zu erzeugen.

Beispiele

@ZUFALLSZAHL = 0,419501 oder ein beliebiger Wert zwischen 0 und 1.

@ZUFALLSZAHL*10 = 6,933674 oder ein beliebiger Wert zwischen 0 und 10.

@GANZZAHL(@ZUFALLSZAHL*50)+1 = 49 oder eine beliebige Ganzzahl zwischen 1 und 50.

@RANG

@RANG(*x*; *Bereich*; [*Rangfolge*]) berechnet die relative Größe oder Position eines Wertes in einem Bereich bezogen auf die anderen Werte in dem Bereich.

Argumente

x ist der Wert, dessen Rang bestimmt werden soll.

Bereich ist die Adresse oder der Name eines Bereichs, der Werte enthält. *Bereich* muß *x* enthalten.

Rangfolge ist ein wahlweises Argument, mit dem angegeben wird, in welcher Reihenfolge *x* eingeordnet werden soll.

Rangfolge entspricht einem der folgenden Werte:

| <u>Rangfolge</u> | <u>1-2-3 behandelt Werte in Bereich so, als wären sie in folgender Folge sortiert</u> |
|------------------|--|
| 0 | Abfallende Reihenfolge (9 bis 1), bevor <i>x</i> geordnet wird; Vorgabe, wenn Sie das Argument weglassen |
| 1 | Steigende Reihenfolge (1 bis 9), bevor <i>x</i> geordnet wird |

Hinweise

1-2-3 weist doppelten Zahlen in *Bereich* dieselbe Rangfolge zu. Doppelte Zahlen wirken sich auf die Rangfolge von nachfolgenden Zahlen in *Bereich* aus. Bei den Werten 2, 4, 6, 8, 8, 10, 12 erscheint die Zahl 8 zweimal und hat eine steigende Rangfolge von 4. Die Zahl 10 hat eine steigende Rangfolge von 6; keine der beiden Zahlen hat eine Rangfolge von 5.

Beispiele

Der Bereich namens UMSATZ (A1..A5) enthält folgende Werte:

5.000 DM
4.000 DM
5.150 DM
4.800 DM
4.900 DM

@RANG(4900;UMSATZ) = 3; 4.900 DM ist der höchste Wert in dem Bereich UMSATZ. Kein Wert hat die Rangfolge 4.

@RANG(4900;UMSATZ;1) = 2; da UMSATZ in steigender Reihenfolge sortiert ist, ist 4.900 DM der zweitniedrigste Wert in dem Bereich UMSATZ. Kein Wert hat die Rangfolge 3.

@ZINS

@ZINS(*Zukwert*; *Aktwert*; *Perioden*) berechnet den periodischen Zinssatz für eine Investition (*Aktwert*), die auf einen zukünftigen Wert *Zukwert* über die angegebene Anzahl von *Perioden* anwachsen soll.

Argumente

Zukwert, *Aktwert* und *Perioden* sind Werte.

Beispiele

Sie haben 10.000 DM in einer Obligation angelegt, die in einem Zeitraum von 5 Jahren auf 18.000 DM anwachsen soll. Der Zinssatz wird monatlich bezahlt. Nun möchten Sie den monatlichen Zins für diese Investition berechnen.

@ZINS(18000;10000;60) ergibt 0,984%, den periodischen (monatlichen) Zinssatz. Um den Jahreszinssatz zu berechnen, benutzen Sie die Formel $((1+\text{@ZINS}(18000;10000;60))^{12}-1)$. Dies ergibt einen jährlichen Zinssatz von 12,47%.

Ähnliche @Funktionen

@EFFBELA berechnet den periodischen Zinssatz, der erforderlich ist, damit eine Rente auf einen bestimmten zukünftigen Wert anwächst.

@REFKONVERT

@REFKONVERT(*Angabe*) wandelt die 1-2-3 Spalten- oder Arbeitsblattbuchstaben A bis IV in Zahlen von 1 bis 256 und Zahlen von 1 bis 256 in die entsprechenden Spalten- oder Arbeitsblattbuchstaben um.

Argumente

Angabe gibt eine 1-2-3 Spalte bzw. ein 1-2-3 Arbeitsblatt an und kann entweder ein Buchstabe von A bis IV, als Text eingegeben, oder eine Ganzzahl von 1 bis 256 sein.

Bei @REFKONVERT spielt die Groß- oder Kleinschreibung keine Rolle; Sie können *Angabe* als Großbuchstabe oder Kleinbuchstabe eingeben.

Beispiele

@REFKONVERT(10) = J

@REFKONVERT("J") = 10

Ähnliche @Funktionen

@SPALTEN zählt die Spalten in einem Bereich und @BLÄTTER die Arbeitsblätter in einem Bereich. @KOORD erstellt eine Zelladresse anhand der von Ihnen angegebenen Werte.

@REGRESSION

@REGRESSION(*X-Bereich*; *Y-Bereich*; *Attribut*; [*berechnen*]) führt eine lineare Mehrfachregression aus und gibt die angegebene Statistik zurück.

Argumente

X-Bereich enthält die unabhängigen Variablen. *X-Bereich* ist der Name bzw. die Adresse eines Bereichs, der bis zu 75 Spalten und 8.192 Zeilen enthalten kann.

Y-Bereich enthält die Werte für die abhängige Variable. *Y-Bereich* ist der Name oder die Adresse eines einspaltigen Bereichs mit derselben Anzahl von Zeilen wie *X-Bereich*.

Attribut gibt an, welcher Regressions-Ausgabewert berechnet werden soll. *Attribut* entspricht einem der folgenden Werte:

| <u>Attribut</u> | <u>1-2-3 berechnet</u> |
|-----------------|---|
| 1 | Konstante |
| 2 | Standardfehler Y |
| 3 | R Quadrat |
| 4 | Anzahl Beobachtungen |
| 5 | Toleranz |
| 101 bis 175 | X-Koeffizient (Steigung) für die unabhängige Variable, die mit <i>Attribut</i> angegeben wird |
| 201 bis 275 | Standardfehler des Koeffizienten für die mit <i>Attribut</i> angegebene unabhängige Variable |

Bei den beiden letzten Attributen numeriert 1-2-3 die unabhängigen Variablen in *X-Bereich*, wobei mit der Zahl 1 begonnen, von oben nach unten in einer Spalte und von links nach rechts gegangen wird.

Ist *X-Bereich* z.B. B2..D7, so benutzen Sie das Attribut 201, um den Standardfehler des Koeffizienten für die unabhängige Variable in Spalte zu ermitteln; benutzen Sie das Attribut 102, um den X-Koeffizienten für die unabhängige Variable in Spalte C zu ermitteln.

Berechnen ist ein wahlweises Argument, mit dem der Schnittpunkt der Y-Achse angegeben wird.

| <u>Berechnen</u> | <u>1-2-3</u> |
|------------------|--|
| 0 | Benutzt 0 als Schnittpunkt der Y-Achse |
| 1 | Berechnet den Schnittpunkt der Y-Achse; Vorgabe, wenn Sie das Argument weglassen |

Hinweise

Bei denselben Daten führen @REGRESSION und Bereich - Regression analysieren zum selben Ergebnis

Beispiele

@REGRESSION

Ähnliche Funktionen

@PROGNOSE(*X*; *Y-Bereich*; *X-Bereich*) ermittelt einen Prognosewert für *X* basierend auf dem linearen Trend zwischen Werten in *Y-Bereich* und *X-Bereich*.

@WIEDERHOLEN

@WIEDERHOLEN(*Zeichenfolge*;n) wiederholt die *Zeichenfolge* n-mal.

Argumente

Zeichenfolge kann in " " (Anführungszeichen) stehender Text, eine Formel, die Text ergibt, oder die Adresse bzw. der Name einer Zelle sein, die ein Label oder eine Formel enthält, die ein Label ergibt.

n kann eine beliebige positive Ganzzahl sein.

Hinweise

Mit WIEDERHOLEN können Sie die Zeichenfolge so oft wiederholen, wie Sie wollen. Die Anzahl von Wiederholungen ist nicht durch die Breite der aktuellen Spalte begrenzt. Hier unterscheidet sich @WIEDERHOLEN von dem sich wiederholenden Justierungszeichen \ (umgekehrter Schrägstrich), mit dem ein Label nur so oft wiederholt wird, bis die aktuelle Spalte gefüllt ist.

Beispiele

@WIEDERHOLEN("Hallo ";3) ergibt Hallo Hallo Hallo.

@WIEDERHOLEN("-";10) ergibt -----.

@ERSETZEN

@ERSETZEN(*Original-Zeichenfolge*; *Startnummer*; *n*; *Ersatz-Zeichenfolge*) ersetzt *n* Zeichen in der *Original-Zeichenfolge* ab der *Startnummer* durch die *Ersatz-Zeichenfolge*.

Argumente

Original-Zeichenfolge und *Ersatz-Zeichenfolge* können Text in " " (Anführungszeichen), Formeln, die Text ergeben, bzw. die Adressen oder Namen von Zellen sein, die Labels oder Formeln enthalten, die Labels ergeben.

Startnummer entspricht dem Zählwert eines Zeichens in *Original-Zeichenfolge*. *Startnummer* kann ein beliebiger positiver Wert oder 0 sein. Wenn *Startnummer* größer ist als die Länge von *Original-Zeichenfolge*, so hängt **@ERSETZEN** die *Ersatz-Zeichenfolge* an die *Original-Zeichenfolge* an.

n kann eine beliebige positive Ganzzahl oder 0 sein. Wenn *n* gleich 0 ist, fügt **@ERSETZEN** *Ersatz-Zeichenfolge* bei *Startnummer* ein, ohne die Zeichen in *Original-Zeichenfolge* zu löschen.

Hinweise

@ERSETZEN zählt alle Zeichen in einer Zeichenfolge, einschließlich Satzzeichen und Leerzeichen. Wenn Sie mit **@ERSETZEN** Text anhängen oder einfügen, denken Sie daran, die erforderlichen Leerzeichen einzufügen.

Benutzen Sie **@FINDEN** mit **@ERSETZEN**, um nach einem Label zu suchen und dieses zu ersetzen oder um eine unbekannte *Startnummer* zu berechnen.

Mit **@ERSETZEN** können Sie eine Zeichenfolge durch eine andere Zeichenfolge ersetzen. So können Sie z.B. in einer Datenbank mit Telefonnummern die Vorwahl austauschen.

Beispiele

@ERSETZEN(ZELLE;**@FINDEN**("-";ZELLE;0);1;"") kopiert das Label in Zelle, d.h. 4-24 als 4/24.

@RECHTS

@RECHTS(*Zeichenfolge*; *n*) gibt die *n* letzten Zeichen in *Zeichenfolge* zurück.

Argumente

Zeichenfolge kann Text in " " (Anführungszeichen), eine Formel, die Text ergibt, bzw. die Adresse oder der Name einer Zelle sein, die ein Label oder eine Formel enthält, die ein Label erzeugt.

n kann eine beliebige positive Ganzzahl oder 0 sein. Ist *n* gleich 0, so ist das Ergebnis eine leere Zeichenfolge. Ist *n* größer als die Länge von *Zeichenfolge*, so gibt @RECHTS die ganze Zeichenfolge zurück.

Hinweise

@RECHTS zählt Satzzeichen und Leerzeichen als Zeichen.

Mit @RECHTS können Sie Teile eines Labels von einer Zelle in eine andere kopieren (z.B. den Nachnamen von Labels extrahieren, die Vor- und Nachnamen enthalten).

In einem Makro können Sie mit @RECHTS Teile von Labels extrahieren, um sie in einer Datenbank, für Unterprogrammaufrufe oder zur Änderung des Makros selbst zu speichern.

Benutzen Sie @RECHTS mit @FINDEN, wenn Sie den genauen Wert für *n* nicht kennen oder wenn *n* variabel ist.

Beispiele

@RECHTS(B3;5) = Umsatz, wenn B3 das Label Januar Umsatz enthält.

Ähnliche @Funktionen

@LINKS gibt die *n* ersten Zeichen in *Zeichenfolge* zurück. @MITTE gibt eine Teilfolge aus *Zeichenfolge* zurück.

@RUNDEN

@ABRUNDEN

@AUFRUNDEN

@RUNDEN($x;n$) rundet den Wert x auf das nächste Vielfache der mit n angegebenen Zehnerpotenz.

@ABRUNDEN($x;n$;*Richtung*) rundet den Wert x auf das nächste Vielfache der mit n angegebenen Zehnerpotenz ab.

@AUFRUNDEN($x;n$;*Richtung*) rundet den Wert x auf das nächste Vielfache der mit n angegebenen Zehnerpotenz auf.

Argumente

x ist ein Wert.

n ist ein Wert von -100 bis 100. Wenn Sie bei @ABRUNDEN und @AUFRUNDEN das Argument n weglassen, benutzt 1-2-3 den Wert 0.

| <u>Ist n</u> | <u>wirkt sich @RUNDEN</u> |
|---------------------------|---|
| Positiv | Auf den Dezimalteil der Zahl (rechts neben der Dezimalstelle) aus. Ist n z.B. 2, so rundet 1-2-3 x auf das nächste Hundertstel. |
| Negativ | Auf den ganzzahligen Teil der Zahl (links neben der Dezimalstelle) aus. Ist n z.B. -2, so rundet 1-2-3 x auf die nächste Hundertstelstelle. |
| 0 | Rundet auf die nächste Ganzzahl. |

Richtung ist ein optionales Argument, mit dem die Art und Weise der Rundung von negativen Werten angegeben wird. *Richtung* kann 0 oder 1 sein.

- Für @AUFRUNDEN: Wenn *Richtung* 0 ist, rundet 1-2-3 negative Werte auf; ist *Richtung* 1, so rundet 1-2-3 negative Werte ab.
- Für @ABRUNDEN: Wenn *Richtung* 0 ist, rundet 1-2-3 negative Werte ab; ist *Richtung* 1, so rundet 1-2-3 negative Werte auf.

Wenn Sie keinen Wert für *Richtung* angeben, verwendet 1-2-3 den Wert 0. Ist x positiv, so hat *Richtung* keine Auswirkungen.

Hinweise

Wählen Sie das feste Zahlenformat , um Werte mit einer bestimmten Anzahl von Dezimalstellen anzuzeigen, wenn 1-2-3 die Werte mit voller Genauigkeit berechnen soll, benutzen Sie @RUNDEN nicht.

Beispiele

@RUNDEN(134,578;2) = 134,58

@RUNDEN(134,578;0) = 135

@RUNDEN(134,578;-2) = 100

@ABRUNDEN(134,578;2) = 134,57

@ABRUNDEN(134,578;0) = 134

@ABRUNDEN(134;578;-2) = 100

@AUFRUNDEN(134,578;2) = 134,58

@AUFRUNDEN(134,578;0) = 135

@AUFRUNDEN(134,578;-2) = 200

Ähnliche @Funktionen

@RUNDENM rundet einen Wert auf ein bestimmtes Vielfaches. @GERADE rundet einen Wert auf die nächste gerade Ganzzahl auf. @UNGERADE rundet einen Wert auf die nächste ungerade Ganzzahl auf. @GANZZAHL schneidet einen Wert ab, wobei der Dezimalteil ignoriert wird. @ABKÜRZEN schneidet einen Wert auf eine bestimmte Dezimalstelle ab.

@RUNDENM

@RUNDENM(*x*; *Faktor*; [*Richtung*]) rundet den Wert *x* auf den nächsten *Faktor*.

Argumente

x und *Faktor* sind beliebige Werte mit demselben Vorzeichen.

Richtung ist ein wahlweises Argument, mit dem angegeben wird, ob *x* auf- oder abgerundet wird.

| Richtung | 1-2-3 rundet x |
|-----------------|---|
| g | |
| 1 | Auf |
| 0 | Zum nächsten Vielfachen; Vorgabe, wenn Sie das Argument weglassen |
| 1 | Ab |

Beispiele

Ein Arbeitsblatt enthält Kostendaten, die auf die nächste Zehnerstelle aufgerundet werden sollen.

@RUNDENM(25,37;0,05;1) = 25,40

@RUNDENM(25,37;0,05;-1) = 25,35

Ähnliche @Funktionen

@GANZZAHL schneidet einen Wert ab, wobei der Dezimalteil ignoriert wird. @RUNDEN, @ABRUNDEN und @AUFRUNDEN runden einen Wert auf eine bestimmte Anzahl von Dezimalstellen. @GERADE rundet einen Wert auf die nächste gerade Ganzzahl auf. @UNGERADE rundet einen Wert auf die nächste ungerade Ganzzahl auf. @ABKÜRZEN schneidet einen Wert auf eine bestimmte Dezimalstelle ab.

@ZEILEN

@ZEILEN(*Bereich*) zählt die Anzahl von Zeilen in *Bereich*.

Argumente

Bereich ist eine Bereichsadresse oder ein Bereichsname.

Hinweise

Benutzen Sie **@ZEILEN** mit {FÜR} in einem Makro, der denselben Schritt mit einer Reihe von Zeilen wiederholt, um festzulegen, wo der Makro anhalten soll.

Beispiele

@ZEILEN(A3..B7) = 5 (Zeilen 3 bis 7).

@ZEILEN(NOTEN) = 43, wenn NOTEN der Bereich B3..B45 ist.

Ähnliche @Funktionen

@SPALTEN zählt die Spalten, und **@BLÄTTER** zählt die Arbeitsblätter in einem Bereich.

Beispiel: @REGRESSION

Sie haben einen Eiskremstand an einem Ausflugsort und möchten in etwa voraussagen, wie viele kg Eiskrem Sie am nächsten Tag verkaufen werden. Sie gehen davon aus, daß die Umsätze von drei Schlüsselfaktoren abhängen: der Anzahl von Sonnenstunden; der Temperatur zu Mittag und der Anzahl von Bussen auf dem nahegelegenen Parkplatz. Nun möchten Sie die Korrelation zwischen diesen Faktoren und Ihrem Umsatz bestimmen. Sie erfassen Daten für einen Zeitraum von sechs Tagen und zeichnen die Beobachtungen in einem Arbeitsblatt auf.

| A | -----A | ----- | B | ----- | C----- | D----- |
|---|---------------|--------------|------------|-----------|-----------|--------|
| 1 | Eiskremumsatz | Sonnenschein | Temperatur | Busse auf | Parkplatz | |
| 2 | 250 | | 3 | 84 | | 10 |
| 3 | 545 | | 5 | 91 | | 7 |
| 4 | 550 | | 5 | 89 | | 8 |
| 5 | 450 | | 6 | 85 | | 10 |
| 6 | 605 | | 6 | 90 | | 11 |
| 7 | 615 | | 7 | 88 | | 9 |

$@REGRESSION(B2..D7;A2..A7;3) = 0,977225$

Da R Quadrat nahe 1 ist, wissen Sie, daß zwischen dem Eiskremumsatz, dem Wetter und der Anzahl von Bussen eine starke Korrelation besteht.

@F

@F(*Bereich*) gibt den Eintrag in der ersten Zelle in *Bereich* als Label zurück.

Argumente

Bereich ist eine Zelladresse oder ein Bereichsname.

Hinweise

@F wird besonders mit Zeichenfolgen-@Funktionen oder Textformeln benutzt, wenn eine Zelle einen Wert enthalten kann und der Eintrag ein Label sein muß (z.B. eine Zelle mit einer Postleitzahl). Benutzen Sie @F, um zu verhindern, daß Textformeln FEHLER ergeben. So gibt z.B. +A1&A2 FEHLER zurück, wenn eine der Zellen einen Wert enthält.

Beispiele

In den Makroinstruktionen

```
{WENN @F(B6)=""}{TON}{ANZEIGE "EINTRAG MUSS EIN LABEL SEIN"}
```

gibt @F eine leere Zeichenfolge zurück, wenn B6 einen Wert enthält oder eine leere Zelle ist. In diesem Fall löst 1-2-3 einen Warnton aus und ändert die Modusanzeige in EINTRAG MUSS EIN LABEL SEIN.

Ähnliche @Funktionen

@W gibt den Eintrag in der ersten Zelle von *Bereich* als Wert zurück. @ISTFOLGE bestimmt, ob eine Zelle ein Label enthält.

@SEC

$@SEC(x)$ berechnet den Sekans des Winkels x . Der Sekans ist das Verhältnis der Hypotenuse zur Ankathete in einem rechtwinkligen Dreieck. Sekans ist der Reziprokwert von Kosinus.

Argumente

x ist ein Winkel gemessen in Radian. x kann ein beliebiger Wert von -2^{63} bis 2^{63} sein.

Beispiele

$@SEC(@GRADINRAD(30)) = 1,154701$, der Sekans eines Winkels von 30 Grad.

Ähnliche @Funktionen

@ARCSEC berechnet den Arkussekans eines Wertes. @ARSECH berechnet den hyperbolischen Arkussekans eines Wertes. @COS berechnet den Kosinus eines Winkels. @SECH berechnet den hyperbolischen Sekans eines Wertes.

@SECH

@SECH(x) berechnet den hyperbolischen Sekans des Winkels x. Der hyperbolische Sekans ist der Reziprokwert des hyperbolischen Kosinus. Das Ergebnis von @SECH ist ein Wert größer als 0 oder kleiner als bzw. gleich 1.

Argumente

x kann ein beliebiger Wert von ca. -709.7827 bis ca. 709.7827 sein.

Beispiele

@SECH(@GRADINRAD(30)) = 0,87701.

Ähnliche @Funktionen

@ARSECH berechnet den hyperbolischen Arkusekans eines Wertes. @SEC berechnet den Sekans eines Wertes.

@SEKUNDE

@SEKUNDE(*Zeitseriennummer*) extrahiert die Sekunden in Form einer Ganzzahl von 0 bis 59 aus *Zeitseriennummer*.

Argumente

Zeitseriennummer ist ein Wert von 0,000000 (Mitternacht) bis 0,999988 (23:59:59).

Beispiele

@SEKUNDE(0,333) = 31.

@SEKUNDE(**@ZEIT**(11;15;45)) = 45, da 45 das *Sekunden*-Argument bei **@ZEIT**(11;15;45) ist.

Ähnliche @Funktionen

@STUNDE extrahiert die Stunden und **@MINUTE** die Minuten aus einer Zeitseriennummer.

@REIHENSUMME

@REIHENSUMME($x;n;m;Koeffizienten$) berechnet die Summe einer Potenzreihe.

Argumente

x ist der Eingabewert der Potenzreihe.

n ist die ursprüngliche Potenz, in die x erhoben werden soll.

m ist das Inkrement, um das n für jeder Faktor in der Reihe erhöht werden muß.

x , n und m sind Werte.

Koeffizienten ist ein Bereich, der die Koeffizienten enthält, mit denen 1-2-3 die einzelnen Potenzen von x multipliziert. Die Anzahl von Zellen in *Koeffizienten* bestimmt die Anzahl von Faktoren in der Reihe. Enthält *Koeffizienten* z.B. zehn Zellen, so enthält die Potenzreihe zehn Faktoren.

Beispiele

Der Bereich DATEN enthält die folgenden Koeffizienten: 0,2, 0,7, 1,3.

@REIHENSUMME(3,5;2;1;DATEN) = 227,5438

@BLÄTTER

@BLÄTTER(*Bereich*) zählt die Anzahl von Arbeitsblättern in *Bereich*.

Argumente

Bereich ist eine Bereichsadresse oder ein Bereichsname.

Hinweise

Benutzen Sie @BLÄTTER mit {FÜR} in einem Makro, der denselben Schritt in einer Reihe von Arbeitsblättern wiederholt, um festzulegen, wo der Makro anhalten soll.

Beispiele

@BLÄTTER(Q_2) = 4, wenn Q_2 dem Bereich B:B3..E:C45 entspricht (den Arbeitsblättern B, C, D und E).

Ähnliche @Funktionen

@SPALTEN zählt die Spalten, und @ZEILEN zählt die Zeilen in einem Bereich. @REFKONVERT setzt die 1-2-3 Spalten- und Arbeitsblattbuchstaben A bis IV in Zahlen von 1 bis 256 um.

@VORZEICHEN

@VORZEICHEN(x) gibt 1 zurück, wenn x ein positiver Wert ist, 0, wenn x gleich 0 ist, und -1, wenn x ein negativer Wert ist.

Argumente

x ist ein beliebiger Wert.

Beispiele

@VORZEICHEN(15) = 1

@VORZEICHEN(15*0) = 0

@VORZEICHEN(-15) = -1

@SIN

@SIN(x) berechnet den Sinus des Winkels x. Der Sinus ist gleich der Gegenkathete/Hypotenuse, wobei Gegenkathete die Seite in einem rechtwinkligen Dreieck ist, die dem spitzen Winkel gegenüberliegt.

Argumente

x ist ein Winkel gemessen in Radian. x kann ein beliebiger Wert von -2^63 bis 2^63 sein.

Beispiele

@SIN(@GRADINRAD(30)) = 0,5, der Sinus eines Winkels von 30 Grad.

Ähnliche @Funktionen

@ARSINH berechnet den hyperbolischen Arkussinus eines Wertes. @ARCSIN berechnet den Arkussinus eines Wertes, um die Spitze eines Winkels zu ermitteln.

@SINH

@SINH(x) berechnet den hyperbolischen Sinus des Winkels x.

Argumente

x kann ein Wert zwischen -709,7827 und 709,7827 sein.

Beispiele

@SINH(@GRADINRAD(30)) = 0,547853.

Ähnliche @Funktionen

@ARSINH berechnet den hyperbolischen Arkussinus eines Wertes. @SIN berechnet den Sinus eines Winkels.

@SCHIEFE

@SCHIEFE(*Bereich*,[*Typ*]) berechnet die Schiefe der Werte in *Bereich*.

Argumente

Bereich ist der Name oder die Adresse eines Bereichs, der Werte enthält. Wenn *Bereich* weniger als drei Werte enthält, so ergibt **@SCHIEFE** den Wert **FEHLER**.

Typ ist ein wahlweises Argument, mit dem angegeben wird, ob die Schiefe für eine Grundgesamtheit oder für Stichproben berechnet werden soll.

| Typ | 1-2-3 berechnet |
|------------|--|
| 0 | Schiefe für eine Grundgesamtheit; Vorgabe, wenn Sie das Argument weglassen |
| 1 | Schiefe für eine Stichprobe |

Hinweise

Mit Schiefe wird die Symmetrie der Verteilung um das Mittel berechnet. Eine positive Schiefe wird mit einer nach links gezogenen Endfläche und eine negative Schiefe mit einer nach rechts gezogenen Endfläche angegeben.

Beispiele

Der Bereich DATEN enthält folgende Werte: 2, 5, 6, 9, 1, 2, 4.

@SCHIEFE(DATEN) = 0,584816

Ähnliche @Funktionen

@KURTOSIS berechnet die Kurtosis der Werte in einer Liste. **@STDABW** und **@PURSTDABW** berechnen die Standardabweichung einer Grundgesamtheit. **@VAR** und **@PURVAR** berechnen die Varianz einer Grundgesamtheit.

@AFALIN

@AFALIN(*Kosten*; *Restwert*; *Lebensdauer*) berechnet die lineare Abschreibung eines Anlagegutes mit dem Anfangswert *Kosten*, einer bestimmten *Lebensdauer* und einem *Restwert* für eine Periode.

Argumente

Kosten entspricht dem Betrag, der für das Anlagegut bezahlt wurde. Bei *Kosten* kann es sich um einen beliebigen Wert handeln.

Restwert entspricht dem Wert des Anlagegutes nach Ablauf der Lebensdauer. *Restwert* kann ein beliebiger Wert sein.

Lebensdauer entspricht der Anzahl von Zeiträumen bis zur Abschreibung eines Anlagegutes auf den Restwert. *Lebensdauer* kann ein beliebiger Wert mit Ausnahme von 0 sein.

Hinweise

Bei der linearen Abschreibung werden die abschreibungsfähigen Kosten (die tatsächlichen Kosten minus dem Restwert) gleichmäßig über die Lebensdauer des Anlagegutes verteilt. Die Lebensdauer entspricht der Anzahl von Perioden (normalerweise Jahre), über die das Anlagegut abgeschrieben wird.

Beispiele

Sie haben eine Büromaschine im Wert von 10.000 DM gekauft. Die Lebensdauer für die Maschine wurde auf 10 Jahre festgesetzt, und der Restwert nach Ablauf von 10 Jahren beläuft sich auf 1.200 DM. Nun möchten Sie den jährlichen Abschreibungsaufwand mit der linearen Abschreibungsmethode berechnen. **@AFALIN**(10000;1200;10) = 880 DM.

Ähnliche @Funktionen

@AFADEG berechnet die Abschreibung mit der festen degressiven Abschreibungsmethode. **@AFADEG** berechnet die Abschreibung mit der geometrisch-degressiven Abschreibungsmethode, **@AFADEGV** berechnet die Abschreibung mit der variablen degressiven Abschreibungsmethode, und **@AFADIG** berechnet die Abschreibung mit der digitalen Abschreibungsmethode.

@NNIEDRIGST

@NNIEDRIGST(*Bereich*; n) ermittelt den n -niedrigsten Wert in *Bereich*.

Argumente

Bereich ist der Name oder die Adresse eines Bereichs, der Werte enthält.

n ist eine beliebige positive Ganzzahl. Wenn n größer als die Anzahl von Werten in *Bereich* ist, ergibt **@NNIEDRIGST** den Wert NV.

Beispiele

Ein Bereich namens ERGEBNIS enthält die folgenden Testergebnisse: 87, 85, 90, 80, 82, 92, 79, 85, 95, 86.

@NNIEDRIGST(ERGEBNIS;3) ergibt 82, das drittniedrigste Testergebnis.

Ähnliche @Funktionen

@NHÖCHST ermittelt den n -höchsten Wert in einem Bereich. **@MAX** und **@PURMAX** ermitteln den größten Wert in einem Bereich. **@MIN** und **@PURMIN** ermitteln den kleinsten Wert in einem Bereich.

@WURZEL

@WURZEL(x) gibt die positive Quadratwurzel von x zurück.

Argumente

x ist ein positiver Wert.

Beispiele

@WURZEL(B3) = 10, wenn B3 den Wert 100 enthält.

@WURZEL(-2) = FEHLER, da x negativ ist.

@WURZELPI

@WURZELPI(x) berechnet die Quadratwurzeln von x^p .

Argumente

x ist ein beliebiger positiver Wert oder 0.

Beispiele

@WURZELPI(0,5) = 1,253314

@WURZELPI(2) = 2,506628

Ähnliche @Funktionen

@WURZEL berechnet die positive Quadratwurzel eines Wertes. @PI erzeugt den Wert π .

@STDABW, @STDABWP, @PURSTDABW, @PURSTDABWP

@STDABW(*Argument-Liste*) berechnet die Standardabweichung der in *Argument-Liste* angegebenen Grundgesamtheit.

@STDABWP(*Argument-Liste*) berechnet die Standardabweichung einer Stichprobe in *Argument-Liste*.

@PURSTDABW(*Argument-Liste*) berechnet die Standardabweichung der in *Argument-Liste* angegebenen Grundgesamtheit, wobei Zellen mit Labels ignoriert werden.

@PURSTDABWP(*Argument-Liste*) berechnet die Standardabweichung einer Stichprobe in *Argument-Liste*, wobei Zellen mit Labels ignoriert werden.

Argumente

Argument-Liste kann eine der folgenden Angaben in einer beliebigen Kombination enthalten: Zahlen, numerische Formeln und Bereichsadressen bzw. Namen von Bereichen, die Zahlen oder Formeln enthalten. Die einzelnen Elemente in *Argument-Liste* müssen durch Argument-Trennzeichen getrennt werden.

Siehe auch Argumente für statistische @Funktionen.

Hinweise

@STDABW und **@PURSTDABW** berechnen die Standardabweichung einer Grundgesamtheit mit der Methode n . Bei dieser Methode wird davon ausgegangen, daß die Werte in *Argument-Liste* der Grundgesamtheit entsprechen. Wenn *Argument-Liste* nur eine Stichprobe der Grundgesamtheit ist, so ist die Standardabweichung "Biased" aufgrund der Fehler, die durch die Stichprobennahme entstehen können.

@STDABWP und **@PURSTDABWP** berechnen die Standardabweichung einer Stichprobe mit der Stichprobe $n-1$. Bei dieser Methode fällt die Standardabweichung etwas größer aus als bei der Methode n , um Fehler auszugleichen, die bei der Probennahme auftreten können. Eine größere Standardabweichung wird von Fehlern bei der Probennahme nicht beeinflusst und ist deshalb genauer.

Die Standardabweichung von Stichproben entspricht der Quadratwurzel der Varianz aller einzelnen Werte vom Mittel.

Beispiele

@STDABW und @STDABWP

Ähnliche @Funktionen

@DSTDABW und @DSTDABWP berechnen die Standardabweichung der Werte in einem Feld einer Datenbanktabelle. @VAR und @PURVAR berechnen die Varianz einer Grundgesamtheit in *Argument-Liste*. @VARP und @PURVARP berechnen die Varianz von Stichproben in *Argument-Liste*.

@FOLGE

@FOLGE($x;n$) wandelt den Wert x in ein Label um, wobei das mit n angegebene Format benutzt wird.

Argumente

x ist ein Wert.

n ist eine Ganzzahl aus der folgenden Tabelle:

| n gleich | @FOLGE ergibt ein Label mit folgendem Format |
|------------------------------|---|
| 0 bis 116 | <u>Fest</u> mit n Dezimalstellen |
| 1000 bis 1116 | <u>Punkt</u> mit $n-1000$ Dezimalstellen |
| -18 bis -1 | <u>Exponential</u> mit @ABS(n) Ziffern |
| 10001 bis 10512 | <u>Standard</u> max. $n-10000$ Zeichen |

Hinweise

@FOLGE ignoriert Formatierungszeichen, mit denen 1-2-3 den Wert x anzeigt. Dies umfaßt alle Währungssymbole und anderen numerischen Formatierungssymbole, unabhängig davon, ob Sie diese eingeben oder ob 1-2-3 sie erstellt, nachdem Sie ein Zahlenformat ausgewählt haben. Wenn Zelle A7 z.B. den formatierten Wert 45,23 DM enthält, ergibt @FOLGE(A7;2) das Label 45,23.

Beispiele

@FOLGE(203;3) = das Label 203,000.

@FOLGE(1,23587;0) = das Label 1.

@FOLGE(20500;1002) = das Label 20.500,00.

@FOLGE(@PI;5) = das Label 3,1416E+000.

@FOLGE(123456,789;10008) = das Label 1,2E+005

Ähnliche @Funktionen

@WERT wandelt eine als Text eingegebene Zahl in den entsprechenden Wert um.

@ZWISCHENSUMME

@ZWISCHENSUMME(*Argument-Liste*) addiert die Werte in *Argument-Liste*. Benutzen Sie **@ZWISCHENSUMME**, um anzugeben, welche Zellen mit **@ENDSUMME** addiert werden sollen.

Argumente

Argument-Liste kann eine der folgenden Angaben in einer beliebigen Kombination enthalten: Zahlen, numerische Formeln und Adressen bzw. Namen von Bereichen, die Zahlen oder Formeln enthalten. Trennen Sie die einzelnen Elemente in *Argument-Liste* durch Argument-Trennzeichen.

Siehe auch Argumente für statistische @Funktionen.

Beispiele

@ZWISCHENSUMME(B5..B9) ergibt die Summe der Werte in B5..B9.

@ZWISCHENSUMME(UMSATZ;M25..R25) ergibt die Summe der Werte in dem Bereich UMSATZ und dem Bereich M25..R25.

Ähnliche @Funktionen

@SUMME addiert die Werte in einer Argument-Liste. **@SUMMENEGATIV** addiert nur die negativen Werte in einer Liste. **@SUMMEPOSITIV** addiert nur die positiven Werte in einer Liste.

@SUMME

@SUMME(*Argument-Liste*) addiert die Werte in *Argument-Liste*.

Argumente

Argument-Liste kann eine der folgenden Angaben in jeder beliebigen Kombination enthalten: Zahlen, numerische Formeln und Adressen bzw. Namen von Bereichen, die Zahlen oder Formeln enthalten. Trennen Sie die Elemente in *Argument-Liste* durch Argument-Trennzeichen.

Siehe auch Argumente für statistische @Funktionen.

Beispiele

@SUMME(B5..B9) ergibt die Summe der Werte in B5..B9.

@SUMME(UMSATZ;M25..R25) ergibt die Summe der Werte in dem Bereich UMSATZ und dem Bereich M25..R25.

Zugehörige SmartIcons



addiert die Werte in dem hervorgehobenen oder angrenzenden Bereich, wenn Sie unter oder rechts neben dem Bereich leere Zellen aufnehmen.

Ähnliche @Funktionen

@DSUMME berechnet die Summe der Werte, die den von Ihnen angegebenen Kriterien entsprechen. @NSUMME summiert jeden *n*-ten Wert in *Argumentliste*, beginnend bei *Versatz*. @ZWISCHENSUMME addiert die Werte in einer *Argument-Liste* und zeigt, welche Werte @ENDSUMME summieren soll. @SUMMENEGATIV addiert nur die negativen Werte in einer Liste. @SUMMEPOSITIV addiert nur die positiven Werte in einer Liste.

@SUMPROD

@SUMPROD(*Argument-Liste*) multipliziert die Werte in den entsprechenden Zellen in mehreren Bereichen und addiert dann die Produkte.

Argumente

Argument-Liste kann eine beliebige Kombination von Bereichen darstellen, die Werte enthalten und dieselbe Größe und Form haben. Wenn die Werte in *Argument-Liste* nicht dieselbe Größe und Form haben, ergibt @SUMPROD den Wert FEHLER.

Siehe auch Argumente für statistische @Funktionen.

Hinweise

Wenn es sich bei den Bereichen in *Argument-Liste* um Spalten handelt, multipliziert @SUMPROD mit Zeilen. Sind die Bereiche in *Argument-Liste* Zeilen, so multipliziert @SUMPROD mit Spalten. Wenn jeder Bereich in *Argument-Liste* mehr als eine Spalte umfaßt, multipliziert @SUMPROD mit Zeilen.

Beispiele

Dieses Beispiel, das aus einer Immobilien-Datenbanktabelle stammt, benutzt @SUMPROD, um die Gesamtprovisionen aufgrund der Hausverkäufe im Februar zu berechnen.

Argument-Liste enthält zwei Bereiche: VERKAUFT (D4..D6) enthält die Preise, die für drei Häuser bezahlt wurden, und PROV (E4..E6) enthält den Provisionssatz in Prozent, den der Makler auf den Verkaufspreis erhält:

| VERKAUFT | PROV |
|-----------|------|
| 25.000 DM | 0,04 |
| 34.580 DM | 0,05 |
| 77.325 DM | 0,04 |

@SUMPROD(VERKAUFT;PROV) = 5.822 DM, die Gesamtprovisionen (1.000 DM + 1.729 DM + 3.093 DM), die an die Makler für den Verkauf der Häuser bezahlt werden müssen.

Ähnliche @Funktionen

@SUMQUADRAT berechnet die Summe der Quadrate der Werte in einer Argument-Liste. @SUMXMY2 berechnet die Summe der Differenz im Quadrat der Werte in entsprechenden Zellen in zwei Bereichen.

@SUMQUADRAT

@SUMQUADRAT(*Argument-Liste*) berechnet die Summe der Quadrate der Werte in *Argument-Liste*.

Argumente

Argument-Liste kann eine der folgenden Angaben in einer beliebigen Kombination enthalten: Zahlen, numerische Formeln und Adressen bzw. Namen von Bereichen, die Zahlen oder Formeln enthalten. Trennen Sie die Elemente in *Argument-Liste* durch Argument-Trennzeichen.

Siehe auch Argumente für statistische @Funktionen.

Beispiele

@SUMQUADRAT(2;4;6) = 56

Ähnliche @Funktionen

@SUMME addiert die Werte in einer Argument-Liste. @SUMPROD multizipiziert die Werte in entsprechenden Zellen in mehreren Bereichen und addiert dann die Produkte.

@SUMXMY2

@SUMXMY2(*Bereich1*;*Bereich2*) subtrahiert die Werte in *Bereich2* von den entsprechenden Zellen in *Bereich1*, erhebt die Differenz ins Quadrat und addiert dann die Ergebnisse.

Argumente

Bereich1 und *Bereich2* sind Bereiche, die Werte enthalten und dieselbe Größe und Form haben. Wenn *Bereich1* und *Bereich2* nicht dieselbe Größe und Form haben, ergibt **@SUMX2MY2** den Wert FEHLER.

Hinweise

Wenn es sich bei *Bereich1* und *Bereich2* um einspaltige Bereiche handelt, subtrahiert 1-2-3 zeilenweise. Sind *Bereich1* und *Bereich2* mehrspaltige Bereiche, so subtrahiert 1-2-3 spaltenweise.

Beispiele

In dem folgenden Beispiel hat *Bereich1* den Namen DIE (für Dienstag) und *Bereich2* den Namen MIT (für Mittwoch).

| DIE | MIT |
|-----|-----|
| 5 | 3 |
| 4 | 4 |
| 7 | 8 |

@SUMXMY2(DIE;MIT) = 5

Ähnliche @Funktionen

@SUMPROD berechnet die Summe der Produkte der Werte in entsprechenden Zellen in mehreren Bereichen.

@SUMQUADRAT berechnet die Summe der Quadrate der Werte in einer Argument-Liste.

@AFADIG

@AFADIG(*Kosten*; *Restwert*; *Lebensdauer*; *Zeitraum*) berechnet die digitale Abschreibung eines Anlagegutes mit dem Anfangswert *Kosten*, einer bestimmten *Lebensdauer* und einem *Restwert* für einen bestimmten *Zeitraum*.

Argumente

Kosten entspricht dem Betrag, der für das Anlagegut bezahlt wurde. Bei *Kosten* kann es sich um einen beliebigen Wert handeln.

Restwert entspricht dem Wert des Anlagegutes nach Ablauf der *Lebensdauer*. Bei *Restwert* kann es sich um einen beliebigen Wert handeln.

Lebensdauer entspricht der Anzahl von Zeiträumen (normalerweise Jahre) bis zur Abschreibung eines Anlagegutes auf den *Restwert*. Bei *Lebensdauer* kann es sich um einen beliebigen Wert handeln, der größer als oder gleich 1 ist.

Zeitraum entspricht der Periode, die für die Abschreibung berechnet wird. Bei *Zeitraum* kann es sich um einen beliebigen Wert handeln, der größer als oder gleich 1 ist.

Hinweise

Mit der digitalen Abschreibung wird die Abschreibungsrate beschleunigt, so daß in früheren Zeiträumen ein höherer Abschreibungsaufwand entsteht als in späteren. (Allerdings ist der Abschreibungsaufwand nicht so hoch wie bei der geometrisch-degressiven Abschreibung.) Die abschreibungsfähigen Kosten entsprechen den aktuellen Kosten minus dem Restwert.

Verwenden Sie **@AFADIG**, wenn Sie zu Beginn der *Lebensdauer* eines Anlagegutes einen höheren Abschreibungsaufwand benötigen.

Beispiele

Sie haben eine Büromaschine im Wert von 10.000 DM gekauft. Die *Lebensdauer* wurde auf 10 Jahre festgelegt, der *Restwert* beläuft sich auf 1.200 DM. Nun möchten Sie den Abschreibungsaufwand für das fünfte Jahr mit der digitalen Abschreibung berechnen: **@AFADIG**(10000;1200;10;5) = 960 DM.

Ähnliche @Funktionen

@AFADEG berechnet die Abschreibung mit der geometrisch-degressiven Abschreibungsmethode. **@AFADEGV** berechnet die Abschreibung mit der variablen degressiven Abschreibung, **@AFAEDEG** benutzt die feste degressive Abschreibungsmethode und **@AFALIN** die lineare Abschreibungsmethode.

Beispiel: @STDABW und @STDABWP

In dieser Tabelle werden die Größe und das Gewicht von zehn zufällig ausgewählten Testpersonen aufgeführt. Nun möchten Sie die Standardabweichung der Größen bestimmen.

$$@STDABW(A2..A11) = 5,793483$$

Wir gehen von der Annahme aus, dass die Testpersonen willkürlich aus einer großen Gruppe von Testpersonen ausgewählt wurden.

$$@STDABWP(A2..A11) = 6,106868$$

| | A ----- | A ----- | B ----- |
|----|-------------|--------------|---------|
| 1 | GRÖSSE (cm) | GEWICHT (kg) | |
| 2 | 190,50 | 72,73 | |
| 3 | 187,96 | 86,36 | |
| 4 | 175,26 | 68,18 | |
| 5 | 175,26 | 76,37 | |
| 6 | 180,34 | 77,27 | |
| 7 | 180,34 | 72,73 | |
| 8 | 187,96 | 75,00 | |
| 9 | 172,72 | 68,18 | |
| 10 | 177,80 | 70,46 | |
| 11 | 179,07 | 86,36 | |

@TAN

@TAN(x) berechnet den Tangens des Winkels x . Der Tangens entspricht der Gegenkathete/Ankathete, wobei Gegenkathete die Seite in einem rechtwinkligen Dreieck ist, die dem spitzen Winkel gegenüberliegt, und Ankathete die Seite, die am spitzen Winkel anliegt.

Argumente

x ist ein Winkel gemessen in Radian. x ist ein Wert von -2^{63} bis $2^{63} \cdot \pi$.

Beispiele

@TAN(@GRADINRAD(35)) = 0,700208, der Tangens eines Winkels von 35 Grad.

Ähnliche @Funktionen

@ARCTAN berechnet den Arkustangens eines Wertes. @TANH berechnet den hyperbolischen Tangens eines Winkels.

@TANH

@TANH(x) berechnet den hyperbolischen Tangens des Winkels x . Der hyperbolische Tangens ist das Verhältnis des hyperbolischen Sinus zum hyperbolischen Kosinus. Das Ergebnis von **@TANH** ist ein Wert von -1 bis 1.

Argumente

x kann ein Wert von ca. -709.7827 bis ca. 709.7827 sein.

Beispiele

@TANH(@GRADINRAD(30)) = 0,480473

Ähnliche @Funktionen

@ARTANH berechnet den hyperbolischen Arkustangens eines Wertes. **@TAN** berechnet den Tangens eines Winkels.

@TVERTEILUNG

@TVERTEILUNG($x;fg;[Typ];[Endflächen]$) berechnet die Studentsche t -Verteilung.

Argumente

Der Wert, den Sie für x eingeben, hängt von dem Wert ab, den Sie für Typ eingeben.

| <u>Ist Typ</u> | <u>so ist x</u> |
|----------------|--|
| 0 | Der kritische Wert oder die Obergrenze für den Wert der Zufallsvariablen der kumulativen t -Verteilung; hier kann es sich um einen beliebigen Wert handeln; Vorgabe, wenn Sie das Argument weglassen |
| 1 | Eine Wahrscheinlichkeit und ein Wert von 0 bis 1 |

fg (für *Freiheitsgrad*) ist die Anzahl von Freiheitsgraden für die Stichprobe. fg ist eine positive Ganzzahl.

Typ ist ein wahlweises Argument, mit dem angegeben wird, wie 1-2-3 @TVERTEILUNG berechnet.

| <u>Typ</u> | <u>1-2-3 berechnet</u> |
|------------|--|
| 0 | Die Signifikanzebene, die dem kritischen Wert entspricht, x ; Vorgabe, wenn Sie das Argument weglassen |
| 1 | Der kritische Wert, der der Signifikanzebene, x , entspricht |

$Endflächen$ ist ein wahlweises Argument, mit dem die Richtung des t -Tests angegeben wird.

| <u>Endflächen</u> | <u>1-2-3 führt folgenden Test aus</u> |
|-------------------|---|
| 1 | Einen t -Test mit einer Endfläche |
| 2 | Einen t -Test mit zwei Endflächen; Vorgabe, wenn Sie das Argument weglassen |

Sie können ein wahlweises Argument nicht ohne die davorstehenden Argumente benutzen.

Hinweise

@TVERTEILUNG berechnet die kumulative t -Verteilung näherungsweise innerhalb eines Bereichs von $\pm 3 \cdot 10^{-7}$. Wenn @TVERTEILUNG das Ergebnis nach 100 Berechnungsversuchen nicht mit einer Genauigkeit von 0,0000001 berechnen kann, ist das Ergebnis FEHLER.

Die t -Verteilung von Student ist die Verteilung des Verhältnisses der standardisierten Normalverteilung zur Quadratwurzel des Quotienten einer Division der Chi-Quadrat-Verteilung durch die Anzahl von Freiheitsgraden.

Beispiele

@TVERTEILUNG(2,228;10) = 0,05

@TVERTEILUNG(0,05;10;1) = 2,228

Ähnliche @Funktionen

@CHIVERTeilUNG berechnet die Chi-Quadrat-Verteilung. @FVERTeilUNG berechnet die F -Verteilung. @TTEST berechnet die Wahrscheinlichkeit, die mit einem t -Test von Student verknüpft ist.

@ANN, @NPER

@ANN(Zahlungen;Zinssatz;Zukwert) berechnet die Anzahl von *Zahlungsperioden*, mit denen ein zukünftiger Wert *Zukwert* mit einem *Zinssatz* erzielt wird. Jede Zahlung entspricht dem Betrag *Zahlungen*.

@NPER(Zahlungen;Zinssatz;Zukwert;[Typ];[Aktwert]) berechnet die Anzahl von *Zahlungsperioden*, die erforderlich sind, damit eine Reihe von gleichen Zahlungen mit einem bestimmten *Aktwert* auf einen zukünftigen Wert *Zukwert* mit einem periodischen *Zinssatz* anwachsen. **@NPER** berechnet entweder eine nachschüssige oder eine vorschüssige Rente, je nach dem Wert, der für *Typ* angegeben wird.

Argumente

Zahlungen ist der Wert der gleichen Investitionen. *Zahlungen* kann ein beliebiger Wert mit Ausnahme von 0 sein.

Zinssatz ist der periodische Zinssatz. *Zinssatz* kann ein beliebiger Wert größer als -1 sein.

Zukwert ist der zu erzielende Betrag. *Zukwert* kann ein beliebiger Wert sein.

Typ ist ein wahlweises Argument, mit dem angegeben wird, ob eine nachschüssige oder eine vorschüssige Rente berechnet wird.

| <u>Typ</u> | <u>1-2-3 berechnet</u> |
|------------|--|
| 0 | Nachschüssige Rente (Zahlungen erfolgen am Ende einer Periode); Vorgabe, wenn Sie das Argument weglassen |
| 1 | Vorschüssige Rente (Zahlungen erfolgen am Anfang einer Periode) |

Aktwert ist ein wahlweises Argument, mit dem der aktuelle Wert einer Reihe von zukünftigen Zahlungen angegeben wird. *Aktwert* kann ein beliebiger Wert sein. Wenn Sie das Argument *Aktwert* weglassen, benutzt 1-2-3 den Wert 0.

Sie können ein wahlweises Argument nicht ohne die davorstehenden Argumente benutzen.

Hinweise

Sie können berechnen, wie lange es dauert, bis ein Darlehen zurückbezahlt wird, indem Sie **@ANN** mit einem negativen *Zukwert* benutzen. So möchten Sie z.B. wissen, wie lange es dauert, bis ein Darlehen in Höhe von 10.000 DM bei einem jährlichen Zinssatz von 10% zurückbezahlt ist, wenn Sie Zahlungen in Höhe von 1.174 DM pro Jahr leisten. **@ABS(@ANN(1174,6;0,1;-10000))** errechnet 20 Jahre zur Rückzahlung des Darlehens.

Beispiele

Sie zahlen am Ende eines jeden Jahres 2.000 DM auf ein Sparkonto ein. Für das Sparkonto werden 7,5% Zinsen jährlich bezahlt, die jährlich ausgeschüttet werden. Nun möchten Sie feststellen, wie lange es dauert, bis Sie 100.000 DM auf dem Sparkonto haben.

@ANN(2000;0,075;100000) = 21,5 Jahre

Wenn Sie die Zahlungen am Anfang jeden Jahres vornehmen, gilt folgende Formel:

@NPER(2000;0,075;100000;1) = 20,76 Jahre

Ähnliche @Funktionen

@LAUF berechnet die Anzahl von Perioden, bis eine einmalige Investition auf einen zukünftigen Wert anwächst.

@ANN2 berechnet die erforderliche Anzahl von Zahlungsperioden bei gleichbleibenden *Zahlungen* zur Akkumulation von *Zukwert* bei periodischem *Zinssatz* und vorschüssiger Rente.

@ZEIT

@ZEIT(*Stunde;Minuten;Sekunden*) berechnet die Zeitseriennummer für *Stunde*, *Minuten* und *Sekunden*.

Argumente

Stunde ist eine Ganzzahl von 0 (Mitternacht) bis 23 (23:00).

Minuten ist eine Ganzzahl von 0 bis 59.

Sekunden ist eine Ganzzahl von 0 bis 59.

Hinweise

Verwenden Sie ein Zeitformat, um die Zeitseriennummer als die entsprechende Zeitangabe anzuzeigen.

Beispiele

Die Formel **@ZEIT(13;0;0)-@ZEIT(9;15;0)*95*24** berechnet den Betrag, den Sie an einem bestimmten Tag an eine Beratungsfirma bezahlen müssen, indem die Startzeit von der Stoppzeit subtrahiert und das Ergebnis mit einem Stundensatz von 95 DM multipliziert wird.

Ähnliche @Funktionen

@ZEITWERT wandelt Labels in Zeitseriennummern um.

@ZEITWERT

@ZEITWERT(*Zeichenfolge*) berechnet die Zeitseriennummer für die in *Zeichenfolge* angegebene Uhrzeit.

Argumente

Zeichenfolge kann eine Zeichenfolge in " " (Anführungszeichen), eine Formel, die eine Zeichenfolge ergibt, bzw. die Adresse oder der Name einer Zelle sein, die ein Label oder eine Formel enthält, die ein Label ergibt. *Zeichenfolge* muß in einem der vier 1-2-3 Zeitformate angegeben werden.

Hinweise

@ZEITWERT wird benutzt, wenn Sie als Labels eingegebene Zeitangaben zur Benutzung in Berechnungen in Zeitseriennummern umsetzen müssen. **@ZEITWERT** wird besonders mit Daten benutzt, die aus einem anderen Programm importiert wurden, wie z.B. einem Textverarbeitungsprogramm.

Verwenden Sie ein Zeitformat, um die Zeitseriennummer als die entsprechende Zeitangabe anzuzeigen.

Beispiele

@ZEITWERT("08:19:27 AM") = 0,34684.

@ZEITWERT("08:19:27 AM") = 0,34684, ausgegeben als 08:19:27 AM, wenn die Zelle als 11:59:59 AM/PM formatiert ist.

Ähnliche @Funktionen

@ZEIT berechnet die Zeitseriennummer, wenn Sie Stunde, Minuten und Sekunden angeben.

@HEUTE

@HEUTE berechnet die Datumseriennummer, die dem laufenden Datum in Ihrem Computer entspricht.

Hinweise

1-2-3 berechnet @HEUTE immer neu, wenn Sie Ihre Daten neu berechnen.

Verwenden Sie ein Datumsformat, um die Datumseriennummer als das entsprechende Datum anzuzeigen.

Beispiele

@HEUTE = 31048 am 1. Januar 1985.

@HEUTE = 33418 am 29. Juni 1991.

@KOMPR

@KOMPR(*Zeichenfolge*) löscht führende, abschließende und aufeinanderfolgende Leerzeichen aus *Zeichenfolge*.

Argumente

Zeichenfolge kann Text in " " (Anführungszeichen), eine Formel, die eine Zeichenfolge ergibt, bzw. die Adresse oder der Name einer Zelle sein, die ein Label oder eine Formel enthält, die ein Label ergibt.

Hinweise

Benutzen Sie @KOMPR, um sicherzustellen, daß Datenbankeinträge keine unnötigen Leerzeichen enthalten, die beim Sortieren eines Bereichs die Sortierfolge beeinflussen würden.

Beispiele

@KOMPR(" 45 3/8") = 45 3/8, wobei die führende Leerstelle vor 45 und eine der beiden Leerstellen zwischen 45 und 3/8 gelöscht wird.

@KOMPR(" 7000 Stuttgart 40") = 7000 Stuttgart 40, wobei die führende Leerstelle vor 7000, zwei der drei Leerstellen zwischen 7000 und Stuttgart sowie eine der beiden Leerstellen zwischen Stuttgart und 40 gelöscht werden.

Ähnliche @Funktionen

@SETZEFOLGE ergibt eine Zeichenfolge, die mit einer bestimmten Anzahl von Leerstellen ausgerichtet ist.

@WAHR

@WAHR gibt den logischen Wert 1 (wahr) zurück.

Hinweise

Wenn eine logische Anweisung wie z.B. $A1=B1$ wahr ist, ist der logische Wert 1. Ist sie falsch, so ist der logische Wert 0.

Die Benutzung von @WAHR ist gleichbedeutend mit der Benutzung des Wertes 1 in Formeln, mit denen logische Bedingungen ausgewertet werden. Mit @WAHR ist die Formel jedoch verständlicher.

Beispiele

@WENN(A6>500;@WAHR;@FALSCH) = 1, wenn Zelle A6 einen Wert größer ist 500 enthält.

Ähnliche @Funktionen

@FALSCH ergibt den logischen Wert 0.

@ABKÜRZEN

@ABKÜRZEN(x;[n]) schneidet x auf die mit n angegebene Anzahl von Dezimalstellen ab.

Argumente

x ist ein Wert.

n ist ein wahlweises Argument und entspricht einem Wert von -100 bis 100.

Wert von n @ABKÜRZEN

| | |
|---------|--|
| Positiv | Wirkt sich auf den Dezimalteil der Zahl (rechts neben der Dezimalstelle) aus. Ist n z.B. 2, so schneidet 1-2-3 x auf das nächste Hundertstel ab. |
| Negativ | Wirkt sich auf den ganzzahligen Teil der Zahl (links neben der Dezimalstelle) aus. Ist n z.B. -2, so schneidet 1-2-3 x auf die nächste Hunderterstelle ab. |
| 0 | schneidet x auf die nächste Ganzzahl ab; Vorgabe, wenn Sie das Argument weglassen |

Hinweise

Wählen Sie das feste Zahlenformat, um Werte mit einer bestimmten Anzahl von Dezimalstellen anzuzeigen, wenn 1-2-3 die Werte mit der vollen Genauigkeit berechnen soll. In diesem Fall benutzen Sie @ABKÜRZEN nicht.

Beispiele

@ABKÜRZEN(123,45) = 123

@ABKÜRZEN(-123,45) = -123

@ABKÜRZEN(123,45;-2) = 100

@ABKÜRZEN(123,45;1) = 123,4

@ABKÜRZEN(-123,45;-2) = -100

@ABKÜRZEN(-123,45;1) = -123,4

Ähnliche @Funktionen

@RUNDEN, @ABRUNDEN und @AUFRUNDEN runden einen Wert auf die angegebene Anzahl von Dezimalstellen.

@RUNDENM rundet einen Wert auf ein angegebenes Vielfaches. @GERADE rundet einen Wert auf die nächste gerade Ganzzahl auf. @UNGERADE rundet einen Wert auf die nächste ungerade Ganzzahl auf. @GANZZAHL schneidet einen Wert ab, wobei der Dezimalteil ignoriert wird.

@GROSS

@GROSS(*Zeichenfolge*) wandelt alle Buchstaben in *Zeichenfolge* in Großbuchstaben um.

Argumente

Zeichenfolge kann ein Text in " " (Anführungszeichen), eine Formel, die Text ergibt, bzw. die Adresse oder der Name einer Zelle sein, die ein Label oder eine Formel enthält, die ein Label ergibt.

Hinweise

Die Groß- bzw. Kleinschreibung wirkt sich auf die Sortierfolge der Labels aus, wenn Sie einen Bereich sortieren. Zwei ansonsten identische Labels stehen möglicherweise nicht hintereinander, wenn sie sich in der Schreibweise des Anfangsbuchstabens unterscheiden.

Beispiele

@GROSS("Kontonummer") = KONTONUMMER.

@GROSS(B2) = WARNUNG, wenn B2 das Label warnung enthält.

Ähnliche @Funktionen

@KLEIN setzt alle Buchstaben in *Zeichenfolge* in Kleinbuchstaben um. @EIGENNAME gibt alle Wörter in *Zeichenfolge* mit großem Anfangsbuchstaben an.

@WERT

@WERT(*Zeichenfolge*) setzt eine als Text eingegebene Zahl in den entsprechenden Wert um.

Argumente

Zeichenfolge kann Text in " " (Anführungszeichen) oder ein Label sein, das nur Zahlen enthält. *Zeichenfolge* kann wie eine Standardzahl (456,7), eine Zahl im Exponentialform (4,567E2), eine gemischte Zahl (45 7/8) oder eine formatierte Zahl (45,67 DM) aussehen.

Hinweise

@WERT ignoriert führende und abschließende Leerstellen; allerdings ergibt @WERT FEHLER, wenn *Zeichenfolge* Leerstellen enthält, mit denen Symbole von den Zahlen getrennt werden (wie z.B. \$ 32,85 oder £ 56,20).

@WERT ergibt 0, wenn *Zeichenfolge* eine leere Zelle oder eine leere Zeichenfolge darstellt, und ergibt FEHLER, wenn *Zeichenfolge* nicht-numerische Zeichen enthält.

Drücken Sie F2 (EDIT) und danach F9 (KALK), um @WERT durch den entsprechenden Wert zu ersetzen.

Sie können nicht mit einem *Zeichenfolgen*-Argument in @WERT rechnen, können jedoch eine Formel mit verschiedenen @WERT-Funktionen erstellen. Zum Beispiel @WERT("22"+"20") = 0, jedoch @WERT("22")+@WERT("20") = 42.

Beispiele

@WERT("543") = der Wert 543.

@WERT(B3) = der Wert 49,75, wenn Zelle B3 das Label 49 3/4 enthält.

@WERT("85%") = der Wert 0,85.

Ähnliche @Funktionen

@FOLGE wandelt einen Wert in ein Label um.

@VAR, @VARP, @PURVAR, @PURVARP

@VAR(*Argument-Liste*) berechnet die Varianz einer Grundgesamtheit in *Argument-Liste*.

@VARP(*Argument-Liste*) berechnet die Varianz einer Stichprobe in *Argument-Liste*.

@PURVAR(*Argument-Liste*) berechnet die Varianz einer Grundgesamtheit in *Argument-Liste*, wobei Zellen mit Labels ignoriert werden.

@PURVARP(*Argument-Liste*) berechnet die Varianz einer Stichprobe in *Argument-Liste*, wobei Zellen mit Labels ignoriert werden.

Argumente

Argument-Liste kann eine der folgenden Angaben in jeder beliebigen Kombination enthalten: Zahlen, numerische Formeln und Adressen bzw. Namen von Bereichen, die Zahlen oder Formeln enthalten. Trennen Sie die einzelnen Elemente von *Argument-Liste* durch Argument-Trennzeichen.

Siehe auch Argumente für statistische @Funktionen.

Hinweise

Die Varianz-@Funktionen werden besonders dann benutzt, wenn Sie statistische Varianzanalysen durchführen müssen.

@VAR und **@PURVAR** berechnen die Varianz mit der Methode n . Bei dieser Methode wird davon ausgegangen, daß die ausgewählten Werte der Grundgesamtheit entsprechen. Stellen die Werte nur eine Stichprobe der Grundgesamtheit dar, so ist die Varianz "Biased" aufgrund der Fehler, die durch die Stichprobennahme auftreten können.

@VARP und **@PURVARP** berechnen die Varianz mit der Methode $n-1$. Bei dieser Methode fällt die Varianz etwas größer aus als bei der Methode n , um Fehler bei der Stichprobennahme zu kompensieren. Eine größere Varianz ist "Unbiased" durch Fehler bei der Probennahme und somit genauer.

Beispiele

@VAR und @VARP

Ähnliche @Funktionen

@DVAR und @DVARP berechnen die Varianz einer Grundgesamtheit von Werten, die den von Ihnen angegebenen Kriterien entsprechen.

@AFADEGV

@AFADEGV(*Kosten*; *Restwert*; *Lebensdauer*; *Beginn*; *Ende*; [*Abschreibung*]; [*Wechsel*]) berechnet die Abschreibung eines Anlagegutes mit dem Anfangswert *Kosten*, einer bestimmten *Lebensdauer* und einem *Restwert* für einen bestimmten Zeitraum, der von *Beginn* bis *Ende* reicht, mit der variablen degressiven Abschreibungsmethode.

Argumente

Kosten entspricht dem Betrag, der für das Anlagegut bezahlt wurde. *Kosten* kann ein beliebiger Wert sein, der größer ist als *Restwert*.

Restwert entspricht dem Wert des Anlagegutes nach Ablauf der Lebensdauer. Bei *Restwert* kann es sich um einen beliebigen Wert handeln.

Lebensdauer entspricht der Anzahl von Zeiträumen bis zur Abschreibung eines Anlagegutes auf den Restwert. Bei *Lebensdauer* kann es sich um einen beliebigen Wert handeln, der größer ist als 0.

Beginn entspricht dem Zeitpunkt, zu dem Sie mit der Abschreibung beginnen möchten. *Beginn* kann ein beliebiger Wert sein, der größer oder gleich 0 ist, jedoch nicht größer als *Lebensdauer*.

Ende entspricht dem Zeitpunkt, zu dem Sie die Abschreibung beenden. *Ende* kann ein beliebiger Wert sein, der größer ist als *Beginn*.

Beginn und *Ende* entsprechen der Lebensdauer des Anlagegutes relativ zum Finanzjahr. Wenn Sie z.B. für das erste Jahr die Abschreibungsrate eines Anlagegutes berechnen möchten, das Sie zu Beginn des zweiten Quartals des Finanzjahres erworben haben, ist *Beginn* 0 und *Ende* 0,75 (1 minus 1/4 eines Jahres). Sie können mit @AFADEGV die Abschreibung für mehrere Perioden berechnen. Mit @AFADEGV haben Sie die Möglichkeit, eine Anfangsperiode zu verwenden, damit die Abschreibung genau zu dem Zeitpunkt beginnen kann, zu dem das Anlagegut in Gebrauch genommen wird. @AFADEGV verwendet die Bruchteile von *Beginn* und *Ende*, um die Anfangsperiode festzulegen. Wenn sowohl *Beginn* als auch *Ende* einen Bruchteil haben, verwendet @AFADEGV den Bruchteil von *Beginn*.

Abschreibung ist ein wahlweises Argument, mit dem Sie den Prozentsatz der linearen Abschreibung angeben, den Sie als Abschreibungssatz benutzen möchten. Wenn Sie keinen Abschreibungssatz angeben, benutzt 1-2-3 den Wert 200%, d.h. den Abschreibungssatz für die geometrisch-degressive Abschreibung. *Abschreibung* ist ein beliebiger Wert, der größer als oder gleich 0 ist. Übliche Abschreibungsrate sind 1,25, 1,50, 1,75 und 2.

Wechsel ist ein wahlweises Argument, das Sie angeben müssen, wenn Sie verhindern wollen, daß @AFADEGV automatisch für die restliche Lebensdauer zur linearen Abschreibung wechselt. Normalerweise wechselt die degressive Abschreibung zur linearen Abschreibung, wenn die Abschreibungsrate größer wird als bei der degressiven Abschreibung.

| <u>Wenn Wechsel</u> | <u>Dann wechselt @AFADEGV</u> |
|---------------------|---|
| 0 | Automatisch zur linearen Abschreibung, wenn bei der linearen Abschreibung die Abschreibungsrate größer ist als bei der degressiven Abschreibung |
| 1 | Niemals zur linearen Abschreibung |

Sie können ein wahlweises Argument nicht ohne die davorstehenden Argumente benutzen.

Hinweise

Die variable degressive Abschreibungsmethode behält einen gleichbleibenden Abschreibungssatz bei, bis der Restwert eines Anlagegutes den Wert der folgenden Gleichung unterschreitet:

$(\text{Buchwert} * ((1 - (\text{Abschreibungssatz} / \text{Lebensdauer})) \text{Lebensdauer}))$

wobei Buchwert = Kosten - Restwert - vorausgegangener Abschreibung ist.

An diesem Punkt wechselt 1-2-3 zur linearen Abschreibung, damit kein großer Restwert bleibt. Beim Wechsel zur linearen Abschreibung korrigiert 1-2-3 gegebenenfalls das Ergebnis von @AFADEGV, um zu gewährleisten, daß die gesamte Abschreibung über die Lebensdauer des Anlagegutes gleich den Kosten des Anlagegutes minus dem Restwert ist.

Beispiele

In diesem Beispiel wird die Abschreibung für eine Büromaschine berechnet, die in der Mitte des ersten Quartals des Finanzjahrs erworben wurde und die 10.000 DM gekostet hat. Die Lebensdauer der Maschine beläuft sich auf 10 Jahre, und der Restwert nach 10 Jahren beträgt 600 DM. Die folgenden Formeln berechnen den Abschreibungsaufwand für jedes der 10 Jahre mit der variablen degressiven Abschreibungsmethode und einer Abschreibungsrate von 150%. Beachten Sie, daß der Wechsel zur linearen Abschreibung im sechsten Jahr erfolgt.

| | | |
|--|---|-----------------|
| @AFADEGV(10000;600;10;0;0.875;1,5) | = | 1.312,50 DM |
| @AFADEGV(10000;600;10;0,875;1,875;1,5) | = | 1.303,13 DM |
| @AFADEGV(10000;600;10;1,875;2,875;1,5) | = | 1.107,66 DM |
| @AFADEGV(10000;600;10;2,875;3,875;1,5) | = | 941,51 DM |
| @AFADEGV(10000;600;10;3,875;4,875;1,5) | = | 800,28 DM |
| @AFADEGV(10000;600;10;4,875;5,875;1,5) | = | 767,79 DM |
| @AFADEGV(10000;600;10;5,875;6,875;1,5) | = | 767,79 DM |
| @AFADEGV(10000;600;10;6,875;7,875;1,5) | = | 767,79 DM |
| @AFADEGV(10000;600;10;7,875;8,875;1,5) | = | 767,79 DM |
| @AFADEGV(10000;600;10;8,875;9,875;1,5) | = | 767,79 DM |
| @AFADEGV(10000;600;10;9,875;10;1,5) | = | <u>95,97 DM</u> |
| Gesamtabschreibung (<i>Kosten</i> minus <i>Restwert</i>) | | 9.400,00 DM |

Ähnliche @Funktionen

@AFADEG berechnet die Abschreibung mit der geometrisch-degressiven Methode. @AFALIN arbeitet mit der linearen Methode und @AFADIG mit der digitalen Methode.

@VVERWEIS

@VVERWEIS(*Argument*; *Spaltenbereich*; *Versatz*) ermittelt den Inhalt der Zelle in einer bestimmten Spalte einer vertikalen Verweistabelle.

Argumente

Argument kann ein Wert oder eine Zeichenfolge sein, je nach Inhalt der ersten Spalte der vertikalen Verweistabelle.

| <u>Erste Spalte</u> | <u>Argument</u> |
|---------------------|--|
| Werte | Ein beliebiger Wert, der größer als oder gleich dem ersten Wert in <i>Spaltenbereich</i> ist. Ist <i>Argument</i> kleiner als der erste Wert in <i>Spaltenbereich</i> , so ergibt @VVERWEIS FEHLER. Ist <i>Argument</i> größer als der letzte Wert in der ersten Spalte des <i>Bereichs</i> , so stoppt @VVERWEIS bei der letzten Zelle in der Spalte, die mit <i>Versatz</i> angegeben wird, und gibt den Inhalt dieser Zelle als Antwort zurück. |
| Labels | Text in " " (Anführungszeichen), eine Formel, die Text ergibt, bzw. die Adresse oder den Namen einer Zelle, die ein Label oder eine Formel enthält, die ein Label ergibt. Entspricht <i>Argument</i> nicht genau dem Inhalt einer Zelle in der ersten Spalte von <i>Bereich</i> , so ergibt @VVERWEIS FEHLER. |

Spaltenbereich entspricht der Zelladresse der vertikalen Verweistabelle. Bei *Spaltenbereich* kann es sich um eine beliebige Bereichsadresse oder einen Bereichsnamen handeln. Ist *Spaltenbereich* ein 3D-Bereich, so benutzt 1-2-3 nur das erste Arbeitsblatt in *Spaltenbereich*.

Versatz entspricht einer Versatzzahl, die der Position entspricht, die die Spalte in *Spaltenbereich* belegt.

Hinweise

@VVERWEIS vergleicht *Argument* mit jeder Zelle in der ersten Spalte der Tabelle. Wenn 1-2-3 eine Zelle in der ersten Spalte findet, die *Argument* enthält (oder wenn *Argument* ein Wert ist den Wert, der *Argument* am nächsten kommt jedoch nicht größer ist als *Argument*), geht 1-2-3 in dieser Zeile um die Anzahl von Spalten nach rechts, die mit *Versatz* angegeben wird. Der Inhalt dieser Zelle wird dann als Antwort zurückgegeben.

Beispiele

@VVERWEIS

Ähnliche @Funktionen

@HVERWEIS ermittelt den Inhalt einer Zelle in einer horizontalen Verweistabelle. @INDEX ermittelt den Inhalt einer Zelle, wenn Sie für Spalte und Zeile Versatzzahlen angeben. @WAHL ersetzt eine Verweistabelle, bei der nur eine Zeile erforderlich ist. @VERGLEICH ermittelt die relative Position einer Zelle mit einem bestimmten Inhalt. @XINDEX ermittelt den Inhalt einer Zelle, die mit Spalten-, Zeilen- und Arbeitsblatt-Titeln angegeben wird. @MAXVERWEIS ergibt eine absolute Referenz auf die Zelle, die den größten Wert in einer Liste mit Bereichen enthält. @MINVERWEIS ergibt eine absolute Referenz auf die Zelle, die den kleinsten Wert in einer Liste mit Bereichen enthält.

@WOCHENTAG

@WOCHENTAG(Datum) gibt den Wochentag in Form einer Ganzzahl von 0 (Montag) bis 6 (Sonntag) anhand von *Datumseriennummer* aus.

Argumente

Datumseriennummer ist eine Datumseriennummer.

Beispiele

@WOCHENTAG(**@DATUM**(91;7;3)) = 2, Mittwoch.

Ähnliche @Funktionen

@MONAT berechnet den Monat anhand einer Datumseriennummer. @JAHR berechnet das Jahr anhand einer Datumseriennummer.

Beispiel:@VAR und @VARP

In dieser Tabelle werden Größe und Gewicht von zehn zufällig ausgewählten Testpersonen aufgeführt. Nun möchten Sie die Varianz bei den Gewichten bestimmen.

@VAR(B2..B11) = 38,77462

Wir gehen davon aus, daß die Testpersonen wahllos aus einer größeren Gruppe von Testpersonen ausgewählt wurden.

@VARP(B2..B11) = 43,08292

| A | ----- | A | ----- | B | -- |
|----|-------|-------------|-------|--------------|----|
| 1 | | GRÖSSE (cm) | | GEWICHT (kg) | |
| 2 | | 190,50 | | 72,73 | |
| 3 | | 187,96 | | 86,36 | |
| 4 | | 175,26 | | 68,18 | |
| 5 | | 175,26 | | 76,37 | |
| 6 | | 180,34 | | 77,27 | |
| 7 | | 180,34 | | 72,73 | |
| 8 | | 187,96 | | 75,00 | |
| 9 | | 172,72 | | 68,18 | |
| 10 | | 177,80 | | 70,46 | |
| 11 | | 179,07 | | 86,36 | |

Beispiel:@VVERWEIS

In einer vertikalen Verweistabelle namens STEUERTABELLE (A3..E11) werden die Steuerbeträge auf der Grundlage der Einkommenshöhe und der Steuerklasse aufgeführt.

@VVERWEIS(35329;BESTEUERBAR;1) in eine Zelle eingegeben, die als Währung ohne Dezimalstellen formatiert ist, ergibt 9.351 DM, die Steuern für eine Einkommenszahl, die 35.329 DM am nächsten kommt, jedoch nicht größer ist als dieser Einkommensbetrag.

| A | ---- | A | ----- | B | ----- | C | ----- | D | ----- | E | ---- |
|----|-----------|----|-------|-------------------------------------|-------|----|--------|----|-------|----|------|
| 1 | | | | ----- S T E U E R K L A S S E ----- | | | | | | | |
| 2 | Einkommen | >= | 1 | | 2 | | 3 | | 4 | | |
| 3 | 35.000 | DM | 9.219 | DM | 7.265 | DM | 11.315 | DM | 8.531 | DM | |
| 4 | 35.050 | DM | 9.241 | DM | 7.282 | DM | 11.340 | DM | 8.552 | DM | |
| 5 | 35.100 | DM | 9.263 | DM | 7.299 | DM | 11.363 | DM | 8.573 | DM | |
| 6 | 35.150 | DM | 9.285 | DM | 7.313 | DM | 11.386 | DM | 8.594 | DM | |
| 7 | 35.200 | DM | 9.307 | DM | 7.330 | DM | 11.411 | DM | 8.615 | DM | |
| 8 | 35.250 | DM | 9.329 | DM | 7.347 | DM | 11.436 | DM | 8.636 | DM | |
| 9 | 35.300 | DM | 9.351 | DM | 7.361 | DM | 11.459 | DM | 8.657 | DM | |
| 10 | 35.350 | DM | 9.373 | DM | 7.377 | DM | 11.483 | DM | 8.678 | DM | |
| 11 | 35.400 | DM | 9.395 | DM | 7.393 | DM | 11.507 | DM | 8.699 | DM | |

@JAHR

@JAHR(Datumseriennummer[Typ]) gibt das Jahr in Form einer Ganzzahl von 0 (1900) bis 199 (2099) aus Datumseriennummer aus.

Argumente

Datumseriennummer ist eine Ganzzahl oder der Name einer Zelle, die eine Ganzzahl enthält, von 1 (1. Januar 1900) bis 73050 (31. Dezember 2099).

Typ ist 0 oder 1. Wenn Typ gleich 1 ist, gibt @JAHR das Jahr als vierstellige Zahl zurück. Wenn Typ 0 ist oder weggelassen wird, gibt @JAHR den Versatz des Jahres gerechnet von 1900 zurück (z.B. 123, um das Jahr 2023 darzustellen).

Hinweise

Addieren Sie 1900 zu @JAHR, um eine vierstellige Jahreszahl zu erhalten. So erzeugt z.B. @JAHR(20181)+1900 das vierstellige Jahr 1955. @JAHR kann das Jahres-Argument für andere Datum-@Funktionen liefern, die auf vorher berechneten Datumsangaben beruhen.

Beispiele

@JAHR(20181) = 55, da die Datumseriennummer 20181 dem 2. April 55 entspricht.

@JAHR(@JETZT) = das laufende Jahr.

@JAHR(@DATUMWERT("14.Feb.92")) = 92.

Ähnliche @Funktionen

@TAG gibt den Tag des Monats (1 bis 31) und @MONAT den Monat (1 bis 12) anhand einer Datumseriennummer wieder.

@RENDITE

@RENDITE(*Valutierungstermin*; *Fälligkeit*; *Coupon*; *Kurs*; [*Rückzahlung*]; [*Zinstermine*]; [*Zinstage*]) berechnet die Rendite für Anleihen mit periodischen Zinszahlungen.

Argumente

Valutierungstermin ist das Abrechnungsdatum für eine Anleihe. *Valutierungstermin* ist eine Datumseriennummer.

Fälligkeit ist das Fälligkeitsdatum der Anleihe. *Fälligkeit* ist eine Datumseriennummer. Wenn *Fälligkeit* kleiner als oder gleich *Valutierungstermin* ist, ergibt @RENDITE den Wert FEHLER.

Coupon ist der jährliche Zinssatz für die Anleihe. *Coupon* ist ein beliebiger positiver Wert oder 0.

Kurs ist der Kurs der Anleihe pro 100 DM Nennwert. *Kurs* ist ein beliebiger positiver Wert.

Rückzahlung ist ein wahlweises Argument, mit dem der Rückzahlungswert der Anleihe pro 100 DM Nennwert angegeben wird. *Rückzahlung* ist ein beliebiger positiver Wert oder 0. Wenn Sie das *Rückzahlung*-Argument weglassen, benutzt 1-2-3 den Wert 100.

Zinstermine ist ein wahlweises Argument, mit dem angegeben wird, wie oft die Coupons pro Jahr bezahlt werden.

| <u>Zinstermine</u> | <u>Anzahl von Zinszahlungen</u> |
|--------------------|--|
| 1 | Jährlich |
| 2 | Halbjährlich; Vorgabe, wenn Sie das Argument weglassen |
| 4 | Vierteljährlich |
| 12 | Monatlich |

Zinstage ist ein wahlweises Argument, mit dem angegeben wird, auf welcher Tagesbasis die Zinsen berechnet werden.

| <u>Zinstage</u> | <u>Tagesbasis</u> |
|-----------------|--|
| 0 | 30/360; Vorgabe, wenn Sie das Argument weglassen |
| 1 | Anzahl/Anzahl |
| 2 | Anzahl/360 |
| 3 | Anzahl/365 |
| 4 | Europäisch 30/360 |

Sie können ein wahlweises Argument nicht ohne die davorstehenden Argumente verwenden.

Beispiele

Eine Anleihe hat einen Valutierungstermin vom 1. Juli 1993 und eine Fälligkeit vom 1. Dezember 1998. Der halbjährliche Coupon beläuft sich auf 5,50%. Die Anleihe wurde zu einem Kurs von 99,50 DM ausgegeben, hat einen Rückzahlungswert von 100 DM, und die Zinstage werden auf der Basis 30/360 berechnet. Nun möchten Sie die Rendite der Anleihe berechnen:

@RENDITE(@DATUM(93;7;1);@DATUM(98;12;1);0,055;99,5;100;2;0) = 0,056072

Ähnliche @Funktionen

@STÜCKZINS berechnet den Stückzins für Anleihen, die periodische Zinsen bezahlen. @KURS berechnet den Kurs pro 100 DM Nennwert für Anleihen, für die periodische Zinsen bezahlt werden. @LAUFZEIT berechnet die jährliche Laufzeit und @MLAUFZEIT die modifizierte jährliche Laufzeit von Anleihen, für die in regelmäßigen Abständen Zinsen bezahlt werden.

@RENDITE2 berechnet die Rendite für Wertpapiere mit periodischen Zinszahlungen auf der Grundlage japanischer Konventionen.

@CHITEST

@CHITEST(*Beobachteter_Bereich*;*[Erwarteter_Bereich]*) führt einen Chi-Quadrat-Test auf Unabhängigkeit der Daten in *Beobachteter_Bereich* oder einen Chi-Quadrat-Test für die Paßgüte mit den Daten in *Beobachteter_Bereich* und *Erwarteter_Bereich* durch.

Argumente

Beobachteter_Bereich und *Erwarteter_Bereich* sind Bereiche derselben Größe. Wenn *Beobachteter_Bereich* und *Erwarteter_Bereich* nicht dieselbe Größe haben oder wenn ein Bereich Labels, Textformeln oder leere Zellen enthält, ergibt **@CHITEST** den Wert FEHLER.

Hinweise

@CHITEST berechnet die Wahrscheinlichkeit, die mit einem Chi-Quadrat-Test verknüpft ist, mit einer Genauigkeit von $\pm 3 \cdot 10^{-7}$.

Beispiele

@CHITEST: Unabhängigkeitstest

@CHITEST: Paßgütetest

Ähnliche @Funktionen

@CHIVERTILUNG berechnet die Chi-Quadrat-Verteilung. **@FTEST** führt einen *F*-Test, **@TTEST** einen Studentischen *t*-Test und **@ZTEST** einen *z*-Test durch.

@KRITBINOMIAL

@KRITBINOMIAL(*Versuche*; *Wahrscheinlichkeit*; *alpha*) gibt die größte Ganzzahl zurück, für die die kumulative Binomialverteilung kleiner als oder gleich *alpha* ist.

Argumente

Versuche entspricht der Anzahl von Bernoulli-Versuchen und kann eine positive Ganzzahl oder 0 sein.

Wahrscheinlichkeit entspricht der Erfolgswahrscheinlichkeit bei einem einzelnen Bernoulli-Versuch und ist ein Wert von 0 bis 1.

alpha entspricht der Kriterienwahrscheinlichkeit und ist ein Wert von 0 bis 1.

Hinweise

@KRITBINOMIAL berechnet die kumulative Binomialverteilung innerhalb einer Genauigkeit von $\pm 3 \cdot 10^{-7}$.

Beispiele

Sie leiten ein kleines Unternehmen, in dem Ölfilter hergestellt werden. Die Filter werden in Posten von 100 hergestellt. Nun gibt es eine 85%ige Wahrscheinlichkeit, daß jeder Filter fehlerfrei ist. Sie möchten 99% sicher sein, daß mindestens eine bestimmte Anzahl von Filtern fehlerfrei ist.

@KRITBINOMIAL(100;0,85;0,01) = 76, die Anzahl von fehlerfreien Filtern

Ähnliche @Funktionen

@BINOMIAL berechnet die binomiale Wahrscheinlichkeits-Massenfunktion bzw. die kumulative Binomialverteilung.

@KOMBIN berechnet den Binomialkoeffizienten. **@PERMUT** berechnet die Anzahl von Permutationen für eine Liste mit Werten.

@DMITTELWERT

@DMITTELWERT(*Datenbank-Bereich*; *Feld*; [*Kriterienbereich*]) berechnet den Mittelwert der Werte in einem *Feld* einer Datenbanktabelle, die den angegebenen *Kriterien* entsprechen.

Argumente

Siehe [Argumente von Datenbank-@Funktionen](#)

Beispiele

[@DMITTELWERT](#)

Ähnliche @Funktionen

[@MITTELWERT](#) und [@PURMITTELWERT](#) berechnen den Mittelwert der Werte in einer Liste.

@DANZAHL, @DPURANZAHL

@DANZAHL(*Datenbank-Bereich;Feld;[Kriterienbereich]*) zählt die nicht leeren Zellen in einem *Feld* einer Datenbanktabelle, die angegebenen *Kriterien* entsprechen.

@DPURANZAHL(*Datenbank-Bereich;Feld;[Kriterienbereich]*) zählt die Zellen, die Werte in einem *Feld* einer Datenbanktabelle enthalten, die bestimmten *Kriterien* entsprechen.

Argumente

Siehe Argumente von Datenbank-@Funktionen

Beispiele

@DANZAHL

Ähnliche @Funktionen

@ANZAHL und @PURANZAHL zählt Zellen in einer Liste mit Bereichen.

@DHOLEN

@DHOLEN(*Datenbank-Bereich;Feld;[Kriterienbereich]*) lädt einen Wert oder ein Label aus einem *Feld* einer Datenbanktabelle, der bzw. das den angegebenen *Kriterien* entspricht.

Argumente

Siehe [Argumente von Datenbank-@Funktionen](#)

Hinweise

Wenn mehr als ein Eintrag den angegebenen *Kriterien* entspricht, ergibt **@DHOLEN** den Wert **FEHLER**.

@DHOLEN wird besonders dann benutzt, wenn Sie einen Wert aus einem einzelnen Datensatz laden müssen, der bestimmten Kriterien entspricht: z.B. die Mitarbeiternummer eines ganz bestimmten Mitarbeiters. Mit **@DHOLEN** können Sie diese Informationen automatisch zur Benutzung in einem Makro, als Argument in einer **@Funktion** oder als Variable in einer Formel laden.

Beispiele

[@DHOLEN](#)

Ähnliche @Funktionen

[@HVERWEIS](#) und [@VVERWEIS](#) ergeben den Inhalt einer bestimmten Zelle in einer horizontalen oder vertikalen Verweistabelle. [@WAHL](#) sucht einen Eintrag in einer Liste. [@INDEX](#) gibt den Inhalt einer Zelle in einer Tabelle aufgrund der relativen Arbeitsblatt-, Spalten- und Zeilenposition wieder. [@XINDEX](#) ergibt den Inhalt einer Zelle in einer Tabelle aufgrund der Arbeitsblatt-, Spalten- und Zeilenüberschriften. [@@](#) gibt den Inhalt einer bestimmten Zelle indirekt wieder.

@DMAX

@DMAX(*Datenbank-Bereich;Feld;[Kriterienbereich]*) ermittelt den größten Wert in einem *Feld* einer Datenbanktabelle, der bestimmten *Kriterien* entspricht.

Argumente

Siehe [Argumente von Datenbank-@Funktionen](#)

Hinweise

Mit **@DMAX** können Sie das jüngste Datum bzw. die jüngste Uhrzeit in einer Liste mit Datums- oder Zeitangaben ermitteln.

Beispiele

[@DMAX](#)

Ähnliche @Funktionen

[@MAX](#) und [@PURMAX](#) ermitteln den größten Wert in einer Liste.

@DMIN

@DMIN(*Datenbank-Bereich*; *Feld*; [*Kriterienbereich*]) ermittelt den kleinsten Wert in einem *Feld* einer Datenbanktabelle, der bestimmten *Kriterien* entspricht.

Argumente

Siehe [Argumente von Datenbank-@Funktionen](#)

Hinweise

Mit **@DMIN** können Sie das früheste Datum bzw. die früheste Uhrzeit in einer Liste mit Datums- oder Zeitangaben finden.

Beispiele

[@DMIN](#)

Ähnliche @Funktionen

[@MIN](#) und [@PURMIN](#) ermitteln den kleinsten Wert in einer Liste.

@DSTDABW, @DSTDABWP

@DSTDABW(*Datenbank-Bereich;Feld;[Kriterienbereich]*) berechnet die Standardabweichung einer Grundgesamtheit in einem *Feld* einer Datenbanktabelle, die bestimmten *Kriterien* entspricht.

@DSTDABWP(*Datenbank-Bereich;Feld;[Kriterienbereich]*) berechnet die Standardabweichung einer Stichprobe der Werte in einem *Feld* einer Datenbanktabelle, die bestimmten *Kriterien* entsprechen.

Argumente

Siehe Argumente von Datenbank-@Funktionen

Hinweise

@DSTDABW benutzt die Methode n (die Methode mit dem systematischen Fehler) zur Berechnung der Standardabweichung einer Grundgesamtheit. Bei der Methode n wird davon ausgegangen, daß die gewählten Werte die ganze Grundgesamtheit darstellen. Wenn die Werte nur eine Stichprobe der Gesamtheit darstellen, ist die Standardabweichung "biased" aufgrund der Fehler bei der Stichprobennahme.

Die Standardabweichung ist die Quadratwurzel der Varianz aller einzelnen Werte vom Mittel.

Beispiele

@DSTDABW und @DSTDABWP

Ähnliche @Funktionen

@STDABW und @PURSTDABW berechnen die Standardabweichung der Grundgesamtheit der Werte in einem Bereich. @STDABWP und @PURSTDABWP berechnen die Standardabweichung von Stichproben. @DVAR berechnet die Varianz einer Grundgesamtheit von Werten, die den angegebenen Kriterien entsprechen.

@DSUMME

@DSUMME(*Datenbank-Bereich*; *Feld*; [*Kriterienbereich*]) berechnet die Summe der Werte in einem *Feld* einer Datenbanktabelle, die bestimmten *Kriterien* entsprechen.

Argumente

Siehe Argumente von Datenbank-@Funktionen

Beispiele

@DSUMME

Ähnliche @Funktionen

@SUMME berechnet die Summe der Werte in einer Liste. @SUMMENEGATIV addiert nur die negativen Werte in einer Liste. @SUMMEPOSITIV addiert nur die positiven Werte in einer Liste.

@DVAR

@DVARP

@DVAR(*Datenbank-Bereich*;Feld;*[Kriterienbereich]*) berechnet die Varianz einer Grundgesamtheit in einem *Feld* einer Datenbanktabelle, die bestimmten *Kriterien* entspricht.

@DVARP(*Datenbank-Bereich*;Feld;*[Kriterienbereich]*) berechnet die Varianz von Stichproben in einem Feld einer Datenbanktabelle, die bestimmten *Kriterien* entsprechen.

Argumente

Siehe [Argumente von Datenbank-@Funktionen](#)

Hinweise

Mit der Varianz wird der Grad der Abweichung einzelner Werte in einer Liste vom Mittelwert aller Werte in der Liste berechnet. Je niedriger die Varianz, desto weniger weichen einzelne Werte vom Mittelwert ab und desto verlässlicher ist der Mittelwert. Eine Varianz von 0 bedeutet, daß alle Werte in der Liste gleich sind. Die Varianz ist bei verschiedenen statistischen Varianzanalyse-Tests erforderlich.

@DVAR berechnet die Varianz mit der Methode n (der Methode mit dem systematischen Fehler). Bei der Methode n wird davon ausgegangen, daß die gewählten Werte die ganze Grundgesamtheit darstellen. Wenn die Werte nur eine Stichprobe der Grundgesamtheit darstellen, ist die Varianz "biased" aufgrund der Fehler bei der Probennahme.

Die Varianz ist das Quadrat der Standardabweichung.

Beispiele

[@DVAR](#) und [@DVARP](#)

Ähnliche @Funktionen

[@VAR](#) und [@PURVAR](#) berechnen die Varianz einer Grundgesamtheit von Werten in einer Liste. [@DSTDABW](#) berechnet die Standardabweichung einer Grundgesamtheit von Werten, die den angegebenen Kriterien entsprechen.

@FTEST

@FTEST(*Bereich1*; *Bereich2*) führt einen *F*-Test aus und gibt die zugehörige Wahrscheinlichkeit zurück.

Argumente

Bereich1 und *Bereich2* sind Bereiche, die die zu testenden Daten enthalten. *Bereich1* und *Bereich2* müssen nicht dieselbe Größe haben.

Hinweise

@FTEST berechnet die mit einem *F*-Test verknüpfte Wahrscheinlichkeit mit einer Genauigkeit von $\pm 3 \cdot 10^{-7}$.

Mit @FTEST ermitteln Sie, ob zwei Stichproben unterschiedliche Varianzen haben.

Beispiele

@FTEST

Ähnliche @Funktionen

@FVERTEILUNG berechnet die *F*-Verteilung. @CHITEST führt einen Chi-Quadrat-Test, @TTEST einen Studentischen *t*-Test und @ZTEST einen *z*-Test durch.

@ENDSUMME

@ENDSUMME(*Argument-Liste*) berechnet die Summe aller Zellen in *Argument-Liste*, die @ZWISCHENSUMME in ihren Formeln enthalten.

Argumente

Argument-Liste kann eine beliebige Kombination von Bereichen sein. Trennen Sie die Elemente von *Argument-Liste* mit Argumenttrennzeichen.

Siehe auch Argumente für statistische @Funktionen

Beispiele

@ENDSUMME

Ähnliche @Funktionen

@SUMME addiert eine Liste mit Werten. @DSUMME addiert Werte in einer Datenbanktabelle, die bestimmten Kriterien gerecht werden. @SUMMENEGATIV addiert nur die negativen Werte in einer Liste. @SUMMEPOSITIV addiert nur die positiven Werte in einer Liste.

@ISTDATEI

@ISTDATEI(*Dateiname*;*[Typ]*) testet, ob *Dateiname* eine Datei im Speicher oder auf Platte ist. Wenn *Dateiname* gefunden wird, ergibt **@ISTDATEI** 1 (wahr); ansonsten ergibt **@ISTDATEI** 0 (falsch).

Argumente

Dateiname ist der volle Name, einschließlich Dateinamenerweiterung, der zu testenden Datei, als Zeichenfolge eingegeben. Wenn 1-2-3 nicht nur im aktuellen Verzeichnis nach der Datei suchen soll, müssen Sie den Pfad als Bestandteil von *Dateiname* angeben.

Typ gibt an, ob im Speicher oder auf Platte nach *Dateiname* gesucht werden soll. Wenn *Typ* gleich 0 ist, sucht 1-2-3 nach *Dateiname* im Speicher; ist *Typ* gleich 1, sucht 1-2-3 nach *Dateiname* auf Platte. Wenn Sie *Typ* weglassen, benutzt 1-2-3 0.

Beispiele

@ISTDATEI("C:\123WBUDGET\KOSTEN_93.123";1) = 1, wenn die Datei KOSTEN_93.123 in C:\123WBUDGET gespeichert ist.

@VERGLEICH

@VERGLEICH(*Zellinhalt*; *Bereich*; [*Typ*]) gibt die Position der Zelle in *Bereich* an, deren Inhalt *Zellinhalt* entspricht.
@VERGLEICH gibt die Position der Zelle als Versatznummer zurück.

Argumente

Zellinhalt kann entweder ein Wert oder eine Zeichenfolge sein. Wenn *Zellinhalt* eine Zeichenfolge ist, können Sie Joker aufnehmen.

Bereich ist ein Bereichsname oder eine Adresse.

Typ ist ein wahlweises Argument, mit dem angegeben wird, wie 1-2-3 *Zellinhalt* mit dem Inhalt der Zellen in *Bereich* vergleicht.

Typ @VERGLEICH gibt die relative Position folgender Zelle zurück

- | | |
|---|---|
| 0 | Erste Zelle, deren Inhalt <i>Zellinhalt</i> entspricht. |
| 1 | Zelle, die den größten Wert enthält, der kleiner als oder gleich <i>Zellinhalt</i> ist; Vorgabe, wenn Sie das Argument weglassen. Sortieren Sie <i>Bereich</i> in steigender Reihenfolge. |
| 2 | Zelle, die den kleinsten Wert enthält, der größer als oder gleich <i>Zellinhalt</i> ist. Sortieren Sie <i>Bereich</i> in fallender Reihenfolge. |

Hinweise

1-2-3 durchsucht *Bereich* von oben nach unten und von links nach rechts in einer Spalte. Wenn Sie einen blattübergreifenden Bereich angeben, durchsucht 1-2-3 das erste Arbeitsblatt in dem Bereich, fährt mit dem zweiten Arbeitsblatt fort usw., bis es eine Übereinstimmung oder das Ende des Bereichs gefunden hat.

Wenn 1-2-3 keine Übereinstimmung für *Zellinhalt* finden kann, ergibt @VERGLEICH FEHLER.

Wenn *Typ* gleich 1 ist und die erste Zelle in *Bereich* einen Wert enthält, der größer als *Zellinhalt* ist, so ergibt @VERGLEICH FEHLER.

Wenn *Typ* gleich 2 ist und die erste Zelle in *Bereich* einen Wert enthält, der kleiner als *Zellinhalt* ist, so ergibt @VERGLEICH FEHLER.

Beispiele

@VERGLEICH

Ähnliche @Funktionen

@HVERWEIS und @VVERWEIS ermitteln den Inhalt von Zellen in horizontalen und vertikalen Verweistabellen.

@INDEX findet den Inhalt einer Zelle, wenn Sie Versatznummern für die Spalte und Zeile angeben. @WAHL ersetzt eine Verweistabelle, bei der nur eine Zeile erforderlich ist. @MAXVERWEIS ergibt eine absolute Referenz auf die Zelle, die den größten Wert in einer Liste mit Bereichen enthält. @MINVERWEIS ergibt eine absolute Referenz auf die Zelle, die den kleinsten Wert in einer Liste mit Bereichen enthält.

@NORMAL

@NORMAL(x;[Mittel];[Stdabw];[Typ]) berechnet die Normal-Verteilungsfunktion für x.

Argumente

x ist die Obergrenze für den Wert der kumulativen Normalverteilung. x ist ein beliebiger Wert; wenn x negativ ist, setzt 1-2-3 es in den absoluten (positiven) Wert um.

Mittel ist ein wahlweises Argument, mit dem das Mittel der Verteilung angegeben wird. Mittel ist ein beliebiger positiver Wert oder 0. Wenn Sie Mittel weglassen, benutzt 1-2-3 0.

Stdabw ist ein wahlweises Argument, mit dem die Standardabweichung der Verteilung angegeben wird. Stdabw ist ein beliebiger positiver Wert oder 0. Wenn Sie Stdabw weglassen, benutzt 1-2-3 1.

Typ ist ein wahlweises Argument, mit dem angegeben wird, welche Funktion @NORMAL berechnen soll.

| <u>Typ</u> | <u>@NORMAL berechnet</u> |
|------------|--|
| 0 | Kumulative Verteilungsfunktion; Vorgabe, wenn Sie das Argument weglassen |
| 1 | Inverse kumulative Verteilung |
| 2 | Wahrscheinlichkeitsdichte-Funktion |

Sie können ein wahlweises Argument nicht ohne die davorstehenden wahlweisen Argumente benutzen.

Hinweise

@NORMAL berechnet die kumulative Verteilungsfunktion mit einer Genauigkeit von $\pm 7.5 \cdot 10^{-8}$ und die inverse kumulative Verteilung mit einer Genauigkeit von $\pm 4.5 \cdot 10^{-4}$

Beispiele

@NORMAL(1,96) = 0,9750

@NORMAL(0,975;0;1;1) = 1,96

@NORMAL(1,96;0;1;2) = 0,058441

Ähnliche @Funktionen

@CHIVERTILUNG berechnet die Chi-Quadrat-Verteilung. @FVERTILUNG berechnet die F-Verteilung.

@POISSON berechnet die Poissonsche Verteilung. @TVERTILUNG berechnet die Studentsche t-Verteilung.

@BEREICHNAME

@BEREICHNAME(*Zelle*) ergibt den Namen des Bereichs, in dem *Zelle* steht.

Argument

Zelle ist eine Zelladresse oder der Name eines einzelligen Bereichs.

Hinweise

Wenn Sie eine Zelle angeben, die in verschiedenen sich überlappenden benannten Bereichen steht, gibt 1-2-3 den ersten Bereichsnamen zurück, den es findet.

Wenn *Zelle* nicht in einem benannten Bereich steht, ergibt @BEREICHNAME FEHLER.

Sie können @BEREICHNAME nur mit Arbeitsmappen im Speicher benutzen.

Beispiele

@BEREICHNAME(A:A2) ergibt UMSATZ, wenn A:A2 in dem Bereich namens UMSATZ steht.

@BEREICHNAME(@ZELLZEIGER("Adresse")) ergibt den Namen des Bereichs, der die aktuelle Zelle enthält.

@SZENARIOLETZT

@SZENARIOLETZT (Dateiname) gibt den Namen der als letztes angezeigten Versionsgruppe in einer Arbeitsmappe während der laufenden Arbeit mit 1-2-3 zurück.

Argumente

Dateiname ist der gesamte, als Text eingegebene Name der Arbeitsmappendatei, einschließlich Erweiterung, nach der Sie suchen möchten. Wenn 1-2-3 die Arbeitsmappe nicht im aktuellen Verzeichnis suchen soll, müssen Sie den Pfad zusammen mit *Dateiname* angeben.

Hinweise

Wenn keine Versionsgruppen in *Dateiname* während der laufenden Arbeit mit 1-2-3 angezeigt wurden, ergibt @SZENARIOLETZT FEHLER.

Beispiele

@SZENARIOLETZT("d:\123r4w\blätter\juli.wk4") ergibt den Namen der zuletzt angezeigten Versionsgruppe in der Arbeitsmappe JULI.WK4, die sich im Verzeichnis D:\123R4W\BLÄTTER befindet.

@SFMITTEL

@SFMITTEL(*Argument-Liste*) berechnet den Standardfehler des Stichprobenmittels für die Werte in *Argument-Liste*.

Argumente

Argument-Liste kann eine der folgenden Angaben in einer beliebigen Kombination enthalten: Zahlen, numerische Formeln und Bereichsadressen oder Bereichsnamen, die Zahlen oder Formeln enthalten. Trennen Sie die Elemente von *Argument-Liste* durch Argumenttrennzeichen.

Beispiele

@SFMITTEL(2;6;8;5;3;9;1;2) = 1,052209

Ähnliche @Funktionen

@GEOMITTEL berechnet das geometrische Mittel der Werte in einer Liste. @HARMITTEL berechnet das harmonische Mittel der Werte in einer Liste. @STDABW und @PURSTDABW berechnen die Standardabweichung der Werte in einer Liste.

@SZENARIOINFO, @VERSIONINFO

@SZENARIOINFO(*Option*;Name;[*Erstellt_Von*]) gibt Informationen über eine Versionsgruppe zurück.

@VERSIONINFO(*Option*;Bereich;Name;[*Erstellt_Von*]) gibt Informationen über eine Version zurück.

Argumente

Option ist eine Zeichenfolge, mit der angegeben wird, welche Informationen 1-2-3 zurückgeben soll.

| <u>Option</u> | <u>1-2-3 gibt folgendes zurück</u> |
|---------------|--|
| Erstellt_Von | Name der Person, die die Version oder die Versionsgruppe erstellt hat. |
| Geändert_Von | Name der Person, die die Version oder die Versionsgruppe zuletzt geändert hat. |
| Erstellt | Datum und Uhrzeit, an der die Version oder die Versionsgruppe erstellt wurde, in Form einer Datum- und Zeitseriennummer. |
| Geändert | Datum und Uhrzeit, bei der die Version oder das Szenarium zuletzt geändert wurde, in Form einer <u>Datum-</u> und <u>Zeitseriennummer</u> . |
| Kommentar | Kommentar für die Version oder die Versionsgruppe; 1-2-3 schneidet den Kommentar ab, wenn er mehr als 512 Zeichen aus jeweils einem Byte umfaßt. |
| Verborgen | 0 (falsch), wenn die Version oder die Versionsgruppe nicht verborgen ist, bzw. 1 (wahr), wenn sie verborgen ist. |
| Geschützt | 0 (falsch), wenn die Version oder die Versionsgruppe nicht geschützt ist, oder 1 (wahr), wenn sie geschützt ist. |

Name ist eine Zeichenfolge, mit der der Name der Version bzw. der Versionsgruppe angegeben wird. Wenn mehr als eine Version oder eine Versionsgruppe denselben *Namen* haben, benutzt 1-2-3 den als letztes erstellten Namen.

Erstellt_Von ist eine Zeichenfolge, mit der der Name des Benutzers angegeben wird, der die Version oder die Versionsgruppe erstellt hat. 1-2-3 benutzt *Erstellt_Von*, um festzustellen, welche Version bzw. welche Versionsgruppe benutzt werden soll.

Bereich ist der Name des Bereichs, in dem die Version steht. *Bereich* muß ein bestehender benannter Bereich sein.

Beispiele

@SZENARIOINFO("Kommentar";"Optimale Lösung";"Kimberly Parker") gibt den Kommentar für die letzte Versionsgruppe namens Optimale Lösung zurück, das von Kimberly Parker erstellt wurde.

@SZENARIOINFO("Erstellt_Von";"Umsatz") gibt den Namen des Benutzers zurück, der die letzte Versionsgruppe namens Umsatz erstellt hat.

@VERSIONINFO("Erstellt_Von";UMSATZBEREICH;"Optimale Lösung";"Anne Parker") gibt das Datum und die Uhrzeit zurück, bei der Anne Parker die letzte Optimale Lösung Version für UMSATZBEREICH erstellt hat.

@VERSIONINFO("Geändert_Von";UMSATZBEREICH;"Schrauben") gibt das Datum und die Uhrzeit zurück, bei der die Version Schrauben in UMSATZBEREICH zuletzt geändert wurde.

@TTEST

@TTEST(*Bereich1*; *Bereich2*; [*Typ*]; [*Endflächen*]) führt einen Studentischen *t*-Test mit den Daten in *Bereich1* und *Bereich2* durch und gibt die zugehörige Wahrscheinlichkeit zurück.

Argumente

Bereich1 und *Bereich2* sind Bereiche, die Werte enthalten. Wenn einer der Bereiche Labels, Textformeln oder leere Zellen enthält, ergibt @TTEST FEHLER.

Typ ist ein wahlweises Argument, mit dem angegeben wird, welche Art von *t*-Test ausgeführt werden soll.

Typ 1-2-3 führt folgenden Test durch

- | | |
|---|--|
| 0 | Einen <i>t</i> -Test für Stichproben aus Grundgesamtheiten mit derselben Varianz; <i>Bereich1</i> und <i>Bereich2</i> müssen nicht dieselbe Anzahl von Zellen enthalten; Vorgabe, wenn Sie das Argument weglassen. |
| 1 | Einen <i>t</i> -Test für Stichproben aus Grundgesamtheiten mit ungleichen Varianzen; <i>Bereich1</i> und <i>Bereich2</i> müssen nicht dieselbe Anzahl von Zellen enthalten. |
| 2 | Einen paarweisen <i>t</i> -Test; <i>Bereich1</i> und <i>Bereich2</i> müssen dieselbe Anzahl von Zellen enthalten. |

Endflächen ist ein wahlweises Argument, mit dem die Richtung des *t*-Tests angegeben wird.

Endflächen 1-2-3 führt folgenden Test durch

- | | |
|---|--|
| 1 | Einen <i>t</i> -Test mit einer Endfläche |
| 2 | Einen <i>t</i> -Test mit zwei Endflächen; Vorgabe, wenn Sie das Argument weglassen |

Sie können ein wahlweises Argument nicht ohne die davorstehenden wahlweisen Argumente benutzen.

Hinweise

@TTEST berechnet die Wahrscheinlichkeit, die mit einem *t*-Test verknüpft ist, mit einer Genauigkeit von $\pm 3 \cdot 10^{-7}$.

Beispiele

@TTEST

Ähnliche @Funktionen

@TVERTEILUNG berechnet die Studentische *t*-Verteilung. @CHITEST führt einen Chi-Quadrat-Test durch, @FTEST einen *F*-Test und @ZTEST einen *z*-Test.

@VERSIONAKTUELL

@VERSIONAKTUELL(*Bereich*) ergibt den Namen der aktuellen Version in *Bereich*.

Argumente

Bereich ist der Name oder die Adresse des Bereichs, für den Sie den Versionsnamen ermitteln möchten.

Hinweise

Wenn keine Version aktuell ist, ergibt @VERSIONAKTUELL FEHLER.

Beispiele

@VERSIONAKTUELL(GEWINNE) ergibt den Namen der aktuellen Version im Bereich GEWINNE.

@VERSIONDATEN

@VERSIONDATEN(*Option*;Zelle;Bereich;Version;[Erstellt_Von]) gibt den Inhalt einer bestimmten Zelle in einer Version zurück.

Argumente

Option ist eine Zeichenfolge, mit der angegeben wird, wie 1-2-3 den Inhalt von *Zelle* zurückgeben soll.

| <u>Option</u> | <u>1-2-3 ergibt</u> |
|---------------|---------------------|
|---------------|---------------------|

| | |
|--------|--|
| Formel | Die Formel in der Zelle als Label oder FEHLER, wenn die Zelle keine Formel enthält |
|--------|--|

| | |
|------|--|
| Wert | Das Ergebnis der Formel, wenn die Zelle eine Formel enthält Den Wert oder das Label, wenn die Zelle einen Wert oder ein Label enthält Eine <u>leere Zeichenfolge</u> , wenn die Zelle leer ist |
|------|--|

Zelle ist der Name oder die Adresse der Zelle, deren Inhalt 1-2-3 zurückgibt. *Zelle* muß in *Bereich* stehen.

Bereich ist der Name des Bereichs, in dem die Version steht. *Bereich* muß ein bestehender benannter Bereich sein.

Version ist eine Zeichenfolge, mit der der Name der Version angegeben wird. Wenn mehr als eine Version denselben Namen hat, benutzt 1-2-3 den als letztes erstellten Namen.

Erstellt_Von ist eine Zeichenfolge, mit der der Name des Benutzers angegeben wird, der die Version erstellt hat. 1-2-3 benutzt *Erstellt_Von*, um festzustellen, welche Version benutzt oder gelöscht werden soll. Wenn *Erstellt_Von* mehrere Versionen mit demselben Namen erstellt hat, benutzt 1-2-3 die als letztes erstellte Version.

Beispiele

@VERSIONDATEN("Formel";A:B12;UMSATZ;"Optimale Lösung") gibt die Formel zurück, die in Zelle A:B12 der als letztes erstellten Version namens Optimale Lösung im Bereich UMSATZ steht.

@VERSIONDATEN("Wert";A:B12;UMSATZ;"Optimale Lösung";"Robin Maier") gibt den Wert oder das Label in Zelle A:B12 der Version namens Optimale Lösung zurück, die als letztes von Robin Maier im Bereich UMSATZ erstellt wurde.

@GEWMITTELWERT

@GEWMITTELWERT(*Datenbereich*; *Gew_Bereich*; [*Typ*]) berechnet den gewichteten Mittelwert der Werte in *Datenbereich*.

Argumente

Datenbereich und *Gew_Bereich* sind die Namen oder Adressen von Bereichen, die Werte enthalten und dieselbe Größe und Form haben.

Wenn *Datenbereich* und *Gew_Bereich* nicht dieselbe Größe und Form haben, ergibt **@GEWMITTELWERT** FEHLER.

Typ ist ein Wert, der festlegt, wie 1-2-3 den gewichteten Mittelwert berechnet.

| <u>Typ</u> | <u>1-2-3 dividiert durch</u> |
|------------|---|
| 0 | Die Summe der Werte in <i>Gew_Bereich</i> ; Vorgabe, wenn Sie das Argument weglassen. |
| 1 | Die Anzahl der Werte in <i>Datenbereich</i> . |

Beispiele

Dieses Beispiel aus einer Datenbanktabelle für Immobilien benutzt **@GEWMITTELWERT** zur Berechnung der gewichteten durchschnittlichen Provision, die für Hausverkäufe im Februar an einen Makler geschuldet werden.

VERKAUFT enthält die Preise für drei Häuser, und PROV enthält die prozentuale Provision der Makler auf den Verkaufspreis:

| VERKAUFT | PROV |
|-----------|------|
| 25.000 DM | 0,04 |
| 34.580 DM | 0,05 |
| 77.325 DM | 0,04 |

@GEWMITTELWERT(VERKAUFT;PROV) = 44.78,62 DM, die gewichtete durchschnittliche Provision, die an einen Makler für den Verkauf von drei Häusern bezahlt werden muß.

Ähnliche @Funktionen

@SUMPROD berechnet die Summe der Produkte der entsprechenden Zellen in mehreren Bereichen.

@ARBEITSTAG

@ARBEITSTAG(*Datum*; *Tage*; [*Freie-Tage*]; [*Wochenenden*]) berechnet die Datumseriennummer, die dem Datum entspricht, das eine bestimmte Anzahl von *Tagen* vor oder nach *Datum* liegt, *Wochenenden* und, wahlweise, *Feiertage* ausgenommen.

Argumente

Datum ist eine Datumseriennummer.

Tage ist eine Ganzzahl. Benutzen Sie eine positive Ganzzahl, um eine Anzahl von Tagen nach *Datum* oder eine negative Zahl, um eine Anzahl von Tagen vor *Datum* anzugeben.

Freie-Tage ist ein wahlweises Argument, mit dem die Feiertage angegeben werden, die aus den Berechnungen von **@ARBEITSTAG** ausgenommen werden sollen. *Freie-Tage* kann eine der folgenden Angaben in einer beliebigen Kombination enthalten: Datumseriennummern, Formeln, deren Auswertung Datumseriennummern ergibt, sowie Bereichsadressen oder Bereichsnamen, die Datumseriennummern enthalten. Trennen Sie die Elemente von *Freie-Tage* durch Argumenttrennzeichen.

Wenn Sie das Argument *Freie-Tage* weglassen, schließt 1-2-3 Feiertage nicht aus der Berechnung von **@ARBEITSTAG** aus.

Wochenenden ist ein wahlweises Argument, mit dem angegeben wird, welche Wochentage als Wochenende betrachtet werden. *Wochenenden* ist eine Zeichenfolge, die die Tage, die Sie als Wochenendtage angeben, mit den Ganzzahlen 0 (Montag) bis 6 (Sonntag) darstellt.

So gibt z.B. "45" an, daß Freitag und Samstag Wochenendtage sind. Wenn Sie *Wochenenden* weglassen, benutzt 1-2-3 "56", d.h. Samstag und Sonntag werden als Wochenendtage betrachtet. Wenn Sie keine Wochenenden angeben möchten, verwenden Sie 7.

Hinweise

Wenn Sie *Wochenenden* verwenden möchten, *Freie-Tage* hingegen nicht, geben Sie für *Freie-Tage* eine leere Zelle an.

Beispiele

Sie möchten das Datum 30 Arbeitstage nach Montag, dem 31. Oktober 1994 feststellen. Der 1. November und der 17. November sollen als Feiertage angegeben werden, so daß Sie für diese Datumsangaben Datumseriennummern in einen Bereich namens FREIE_TAGE eingeben. Samstag und Sonntag sollen als Wochenendtage angegeben werden, so daß Sie das Argument *Wochenenden* weglassen.

@ARBEITSTAG(**@DATUM**(94;10;31);30;FREIE_TAGE) = 34683 oder Donnerstag, der 15. Dezember 1994.

Ähnliche @Funktionen

@DAYS360 und **@D360** berechnen die Anzahl von Tagen zwischen zwei Datumseriennummern.

@NETTOARBEITSTAGE berechnet die Anzahl von Tagen zwischen zwei Datumsangaben, *Wochenenden* und *Feiertage* ausgenommen. **@NÄCHSTMONAT** berechnet das Datum, das eine bestimmte Anzahl von Monaten vor oder nach einem bestimmten Datum liegt.

Beispiel: @CHITEST (Unabhängigkeitstest)

@CHITEST(B3..D5) = 0,080809

| A | A | B | C | |
|---|---------|-----|-----------|----|
| 1 | | Roh | Geglättet | |
| 2 | | | | |
| 3 | Hoch | 42 | 51 | |
| 4 | Mittel | | 25 | 37 |
| 5 | Niedrig | 85 | 68 | |

Beispiel: @CHITEST (Paßgütetest)
@CHITEST(A2..A9;B2..B9) = 0,996882

| A | ----- | A | ---- | B | ----- |
|---|-------------|----|-----------|----|-------|
| 1 | Beobachtung | | Erwartung | | |
| 2 | | 25 | | 23 | |
| 3 | | 34 | | 35 | |
| 4 | | 87 | | 91 | |
| 5 | | 91 | | 88 | |
| 6 | | 34 | | 34 | |
| 7 | | 23 | | 22 | |
| 8 | | 56 | | 60 | |
| 9 | | 70 | | 68 | |

Beispiel: @DMITTELWERT

In einer Datenbanktabelle namens VERKÄUFE werden die Hausverkäufe in München, Stuttgart und Mannheim im April und Mai aufgeführt. Die Verkaufspreise der Häuser werden in dem Feld VERKAUFT aufgeführt. Nun möchten Sie den Mittelwert eines Hauses berechnen, das in Mannheim verkauft wurde:

@DMITTELWERT(VERKÄUFE;"VERKAUFT";STADT="Mannheim") = 365.667 DM

| A | A | B | C |
|----|------------------|-----------|----------|
| 1 | ADRESSE | STADT | VERKAUFT |
| 2 | Kaiserstr. 467 | Mannheim | 720.000 |
| 3 | Bergstr. 183 | München | 318.000 |
| 4 | Liststr. 64 | Stuttgart | 332.000 |
| 5 | Mozartstr. 80 | Mannheim | 278.000 |
| 6 | Karlsallee 14 | Mannheim | 160.000 |
| 7 | Königstr. 116 | Mannheim | 227.000 |
| 8 | Hospitalstr. 130 | München | 397.000 |
| 9 | Lange Str. 12 | München | 303.000 |
| 10 | Alexanderstr. 36 | Mannheim | 669.000 |
| 11 | Olgastr. 234 | Mannheim | 140.000 |

Beispiel: @DANZAHL

Eine Datenbanktabelle namens APR_VERKÄUFE führt die Hausverkäufe im Monat April auf. Die Heizungen der Häuser werden in einem Feld namens HEIZUNG aufgeführt. Nun möchten Sie die Anzahl von Häusern ermitteln, die mit Gas beheizt werden:

@DANZAHL(APR_VERKÄUFE;"HEIZUNG";HEIZUNG="Gas") = 4

| A | ---- | A | ----- | B | ----- | C | ----- | D | ---- |
|---|---------------|---|----------|---------|-------|-------|-------|---|------|
| 1 | ADRESSE | | SCHLAFZ. | HEIZUNG | | PREIS | | | |
| 2 | Kaiserstr.457 | 5 | Gas | 290.000 | DM | | | | |
| 3 | Bergstr. 183 | 5 | Öl | 105.000 | DM | | | | |
| 4 | Liststr. 64 | 2 | Gas | 135.000 | DM | | | | |
| 5 | Karlsallee 14 | 4 | Elektr. | 128.000 | DM | | | | |
| 6 | Königstr. 116 | 2 | Gas | 174.000 | DM | | | | |
| 7 | Lange Str. 12 | 3 | Gas | 195.000 | DM | | | | |

Beispiel: @DHOLEN

In einer Datenbanktabelle namens VERKÄUFE werden die Hausverkäufe in drei Städten im April und Mai aufgeführt. Die Maklerprovisionen werden in dem Feld PROV aufgeführt. Nun möchten Sie die Maklerprovision beim Verkauf des Hauses in Lange Str. 12 ermitteln:

@DHOLEN(VERKÄUFE;"PROV";ADRESSE="Lange Str. 12") = 12.120 DM

| A | ----- | A | ----- | B | ----- | C | -- |
|---|------------------|---|-------|---------|-------|--------|----|
| 1 | ADRESSE | | | MAKLER | | PROV | |
| 2 | Kaiserstr. 467 | | | Maier | | 28.800 | |
| 3 | Bergstr. 183 | | | Müller | | 12.720 | |
| 4 | Liststr. 64 | | | Maier | | 19.920 | |
| 5 | Mozartstr. 80 | | | Schmidt | | 11.120 | |
| 6 | Karlsallee 14 | | | Bauer | | 9.600 | |
| 7 | Königstr. 116 | | | Müller | | 13.620 | |
| 8 | Hospitalstr. 130 | | | Bauer | | 15.880 | |
| 9 | Lange Str. 12 | | | Maier | | 12.120 | |

Beispiel: @DMAX

In einer Datenbanktabelle namens VERKÄUFE werden die Hausverkäufe in München, Stuttgart und Mannheim im April und Mai aufgeführt. Der Verkaufspreis der Häuser wird in dem Feld VERKAUFT aufgeführt. Nun möchten Sie den höchsten Preis ermitteln, der für ein Haus in Mannheim bezahlt wurde:

@DMAX(VERKÄUFE;"VERKAUFT";STADT="Mannheim") = 720.000 DM

| A | ----- | A | ----- | B | ----- | C | ----- |
|----|-------|------------------|-------|-----------|-------|----------|-------|
| 1 | | ADRESSE | | STADT | | VERKAUFT | |
| 2 | | Kaiserstr. 467 | | Mannheim | | 720.000 | |
| 3 | | Bergstr. 183 | | München | | 318.000 | |
| 4 | | Lisztstr. 64 | | Stuttgart | | 800.000 | |
| 5 | | Mozartstr. 80 | | Mannheim | | 278.000 | |
| 6 | | Karlsallee 14 | | Mannheim | | 160.000 | |
| 7 | | Königstr. 116 | | Mannheim | | 227.000 | |
| 8 | | Hospitalstr. 130 | | München | | 397.000 | |
| 9 | | Lange Str. 12 | | München | | 303.000 | |
| 10 | | Alexanderstr. 36 | | Mannheim | | 669.000 | |
| 11 | | Olgastr. 234 | | Mannheim | | 140.000 | |

Beispiel: @DMIN

In einer Datenbanktabelle namens VERKÄUFE werden die Hausverkäufe in München, Stuttgart und Mannheim im April und Mai aufgeführt. Die Verkaufspreise der Häuser werden in dem Feld VERKAUFT aufgeführt. Nun möchten Sie den niedrigsten Preis ermitteln, der für ein Haus in Mannheim bezahlt wurde:

@DMIN(VERKÄUFE;"VERKAUFT";STADT="Mannheim") = 140.000 DM

| A | ----- A ----- | B ----- | C --- |
|----|------------------|-----------|----------|
| 1 | ADRESSE | STADT | VERKAUFT |
| 2 | Kaiserstr. 467 | Mannheim | 720.000 |
| 3 | Bergstr. 183 | München | 318.000 |
| 4 | Lisztstr. 64 | Stuttgart | 332.000 |
| 5 | Mozartstr. 80 | Mannheim | 278.000 |
| 6 | Karlsallee 14 | Mannheim | 160.000 |
| 7 | Königstr. 116 | Mannheim | 227.000 |
| 8 | Hospitalstr. 130 | München | 397.000 |
| 9 | Lange Str. 12 | München | 130.000 |
| 10 | Alexanderstr. 36 | Mannheim | 669.000 |
| 11 | Olgastr. 234 | Mannheim | 140.000 |

Beispiel: @DSTDABW und @DSTDABWP

In dieser Tabelle werden die Größe und das Gewicht von zehn zufällig gewählten Testpersonen aufgeführt. Sie möchten die Standardabweichung der Größe der Personen ermitteln, die mehr als 75 kg wiegen.

@DSTDABW(A1..A11;"GRÖSSE";GEWICHT>75) = 25,59654

Angenommen, bei den zehn Testpersonen handelt es sich um zufällig gewählte Stichproben aus einer größeren Gruppe von Testpersonen.

@DSTDABWP(A1..B11;"GRÖSSE";GEWICHT>75) = 31,99567

| A | ----- | A | ----- | B | ----- |
|----|-------|-------------|-------|--------------|-------|
| 1 | | GRÖSSE (cm) | | GEWICHT (kg) | |
| 2 | | 190,50 | | 72,73 | |
| 3 | | 187,96 | | 86,36 | |
| 4 | | 175,26 | | 68,18 | |
| 5 | | 175,26 | | 76,37 | |
| 6 | | 180,34 | | 77,27 | |
| 7 | | 180,34 | | 72,73 | |
| 8 | | 187,96 | | 75,00 | |
| 9 | | 172,72 | | 68,18 | |
| 10 | | 177,80 | | 70,46 | |
| 11 | | 179,07 | | 86,36 | |

Beispiel: @DSUMME

In einer Datenbanktabelle namens VERKÄUFE werden die Hausverkäufe in München, Stuttgart und Mannheim im April und Mai aufgeführt. Die Maklerprovisionen werden in dem Feld namens PROV aufgeführt. Nun möchten Sie die Gesamtprovisionen berechnen, die der Makler Bauer verdient hat:

@DSUMME(VERKÄUFE;"PROV";MAKLER="Bauer") = 25.480 DM

| A | ----- | A | ----- | B | ----- | C | ---- |
|---|-----------------|---|-------|---------|-------|--------|------|
| 1 | ADRESSE | | | MAKLER | | PROV | |
| 2 | Kaiserstr. 467 | | | Maier | | 28.800 | |
| 3 | Bergstr. 183 | | | Müller | | 12.720 | |
| 4 | Lisztstr. 64 | | | Maier | | 19.920 | |
| 5 | Mozartstr. 80 | | | Schmidt | | 11.120 | |
| 6 | Karlsallee 14 | | | Bauer | | 9.600 | |
| 7 | Königstr. 116 | | | Müller | | 13.620 | |
| 8 | Hospitalstr. 13 | | | Bauer | | 5.880 | |
| 9 | Lange Str. 12 | | | Maier | | 12.120 | |

Beispiel: @DVAR und @DVARP

In dieser Tabelle werden die Größe und das Gewicht von zehn zufällig gewählten Testpersonen aufgeführt. Nun möchten Sie die Varianz des Gewichts der Personen ermitteln, die größer sind als 1,80 m.

$$\text{@DVAR}(B2..B11;"\text{GEWICHT}";\text{GRÖSSE}>180) = 38,77462$$

Angenommen, bei den Testpersonen handelt es sich um zufällig gewählte Stichproben aus einer größeren Gruppe mit Testpersonen.

$$\text{@DVARP}(B2..B11;"\text{GEWICHT}";\text{GRÖSSE}>180) = 43,08292$$

| A | ----- | A | ----- | B | -- |
|----|-------|-------------|-------|--------------|----|
| 1 | | GRÖSSE (cm) | | GEWICHT (kg) | |
| 2 | | 190,50 | | 72,73 | |
| 3 | | 187,96 | | 86,36 | |
| 4 | | 175,26 | | 68,18 | |
| 5 | | 175,26 | | 76,37 | |
| 6 | | 180,34 | | 77,27 | |
| 7 | | 180,34 | | 72,73 | |
| 8 | | 187,96 | | 75,00 | |
| 9 | | 172,72 | | 68,18 | |
| 10 | | 177,80 | | 70,46 | |
| 11 | | 179,07 | | 86,36 | |

Beispiel: @FTEST

@FTEST(A2..A13;B2..B15) = 0,157348

| A | ----- A | ----- B | ----- |
|----|-------------|------------|-------|
| 1 | Stichprobel | Stichprobe | |
| 2 | 84,5 | 1,65 | |
| 3 | 80,7 | 4,58 | |
| 4 | 34,5 | 42,6 | |
| 5 | 54,6 | 4,37 | |
| 6 | 50,5 | 30,8 | |
| 7 | 33,7 | 97,7 | |
| 8 | 46,8 | 87,2 | |
| 9 | 47,6 | 40,7 | |
| 10 | 22,8 | 38,4 | |
| 11 | 15,5 | 10,6 | |
| 12 | 60,6 | 56,3 | |
| 13 | 80,5 | 70,5 | |
| 14 | | 9,04 | |
| 15 | | 97,3 | |

Beispiel: @ENDSUMME

Die Formel @ENDSUMME in Zelle A10 berechnet die Summe aller Zellen in A1..A8, die @ZWISCHENSUMME in ihren Formeln enthalten (A4 und A8).

| A | --- | A | ----- | B | ----- |
|----|-----|----|-------|-------------------------|-------|
| 1 | | 10 | | | |
| 2 | | 15 | | | |
| 3 | | | | | |
| 4 | | 25 | | @ZWISCHENSUMME (A1..A2) | |
| 5 | | | | | |
| 6 | | 20 | | | |
| 7 | | 25 | | | |
| 8 | | 45 | | @ZWISCHENSUMME (A6..A7) | |
| 9 | | | | | |
| 10 | | 70 | | @ENDSUMME (A1..A8) | |

@XINDEX

@XINDEX(*Bereich*; *Spaltentitel*; *Zeilentitel*; [*Arbeitsblatt-Titel*]) gibt den Inhalt einer Zelle zurück, die bei der Schnittstelle zwischen *Spaltentitel*, *Zeilentitel* und (wahlweise) *Arbeitsblatt-Titel* steht.

Argumente

Bereich ist eine Bereichsadresse oder ein Bereichsname.

Spaltentitel ist der Inhalt einer Zelle in der ersten Zeile von *Bereich*.

Zeilentitel ist der Inhalt einer Zelle in der ersten Spalte von *Bereich*.

Arbeitsblatt-Titel ist ein wahlweises Argument, das den Inhalt der ersten Zelle in *Bereich* angibt.

Spaltentitel, *Zeilentitel* und *Arbeitsblatt-Titel* können Werte oder Zeichenfolgen sein.

Beispiele

@XINDEX

Ähnliche @Funktionen

@WAHL findet einen Eintrag in einer Liste. @HVERWEIS und @VVERWEIS finden Einträge in horizontalen und vertikalen Verweistabellen. @VERGLEICH gibt die relative Position einer Zelle in einem Bereich zurück.

@MAXVERWEIS ergibt eine absolute Referenz auf die Zelle, die den größten Wert in einer Liste mit Bereichen enthält. @MINVERWEIS ergibt eine absolute Referenz auf die Zelle, die den kleinsten Wert in einer Liste mit Bereichen enthält.

Beispiel: @VERGLEICH

Die Dosierung eines Medikamentes wird vom Körpergewicht des Patienten bestimmt. Das Körpergewicht des Patienten, eingegeben in eine Zelle PATIENT_GEWICHT beträgt 125 Pfund.

@INDEX(A2..C7;2;@VERGLEICH(PATIENT_GEWICHT;A2..C7;1)) = 2

| A | A | B | C |
|---|-------|-----------|------------------|
| 1 | Pfund | Kilogramm | Anzahl Tabletten |
| 2 | 50 | 22,5 | 1,5 |
| 3 | 100 | 45,5 | 2,0 |
| 4 | 150 | 68,0 | 2,5 |
| 5 | 200 | 90,5 | 3,0 |
| 6 | 250 | 113,5 | 3,5 |
| 7 | 300 | 136,0 | 4,0 |

Beispiel: @TTEST

@TTEST(A2..A13;B2..B13) = 0,050022

| A | ----- A | ----- B | ---- |
|----|-------------|------------|------|
| 1 | Stichprobel | Stichprobe | |
| 2 | 84,5 | 65,1 | |
| 3 | 80,7 | 85,4 | |
| 4 | 34,5 | 62,4 | |
| 5 | 54,6 | 73,4 | |
| 6 | 50,5 | 80,3 | |
| 7 | 33,7 | 66,7 | |
| 8 | 46,8 | 87,2 | |
| 9 | 47,6 | 70,4 | |
| 10 | 22,8 | 30,2 | |
| 11 | 15,5 | 60,1 | |
| 12 | 60,6 | 56,3 | |
| 13 | 80,5 | 70,5 | |

Beispiel: @XINDEX

Eine Tabelle namens SÄTZE (A2..E7) führt die Sätze für die Zustellung von Paketen in verschiedenen Städten auf.

@XINDEX(SÄTZE;"New York";1) = 9,29 DM, der Satz für das Versenden eines Paketes vom Typ 1 nach New York.

@XINDEX(SÄTZE;"Paris";5) = 23,88 DM, der Satz für das Versenden eines Paktes vom Typ 5 nach Paris.

| A | ---- | A | ---- | B | ----- | C | --- | D | ----- | E | ----- |
|---|----------|--------|-------|-----------|-------|----------|-------|---|-------|---|-------|
| 1 | | | | | ----- | ZIELORT | ----- | | | | |
| 2 | Pakettyp | London | Paris | Frankfurt | | New York | | | | | |
| 3 | 1 | 18,36 | 19,33 | 20,12 | | 9,29 | | | | | |
| 4 | 2 | 20,32 | 21,66 | 22,03 | | 11,25 | | | | | |
| 5 | 3 | 22,44 | 23,88 | 24,00 | | 16,85 | | | | | |
| 7 | 5 | 28,32 | 29,00 | 29,80 | | 19,54 | | | | | |

@ZTEST

@ZTEST(*Bereich1*; *Mittel1*; *Stdabw1*; [*Endflächen*]; [*Bereich2*]; [*Mittel2*]; [*Stdabw2*]) führt einen z-Test mit einer oder zwei Grundgesamtheiten durch und gibt die zugehörige Wahrscheinlichkeit zurück.

Argumente

Bereich1 ist ein Bereich, der den ersten oder einzigen zu testenden Datensatz umfaßt.

Mittel1 ist das bekannte Grundgesamtheitsmittel von *Bereich1* und kann ein beliebiger Wert sein.

Stdabw1 ist die bekannte Standardabweichung der Grundgesamtheit von *Bereich1*. *Stdabw1* ist ein Wert größer als 0.

Endflächen ist ein wahlweises Argument, mit dem die Richtung des z-Tests angegeben wird.

| <u>Endfläche</u> <u>n</u> | <u>1-2-3 führt folgenden Test aus</u> |
|------------------------------|--|
| 1 | Einen z-Test mit einer Endfläche |
| 2 | Einen z-Test mit zwei Endflächen; Vorgabe, wenn Sie das Argument weglassen |

Bereich2 ist der Bereich, der den zweiten zu testenden Datensatz enthält.

Mittel2 ist das bekannte Grundgesamtheitsmittel von *Bereich2* und kann ein beliebiger Wert sein. Wenn Sie *Mittel2* weglassen, benutzt 1-2-3 0.

Stdabw2 ist die bekannte Standardabweichung der Grundgesamtheit von *Bereich2*. *Stdabw2* ist ein Wert größer als 0. Wenn Sie *Stdabw2* weglassen, benutzt 1-2-3 1.

Sie können ein wahlweises Argument nicht ohne die davorstehenden Argumente benutzen.

Hinweise

@ZTEST berechnet die Wahrscheinlichkeit, die mit einem z-Test verbunden ist, mit einer Genauigkeit von $\pm 7.5 \cdot 10^{-8}$.

Beispiele

Der Bereich A1..A8 enthält folgende Werte: 12, 19, 21, 22, 18, 16, 15, 17. Wenn das Grundgesamtheitsmittel dieser Werte 17,5 und die Standardabweichung der Grundgesamtheit 3,041381 ist, ist $z = 1,394972$

@ZTEST(A1..A8;16;3,041381;1) = 0,081512

Ähnliche @Funktionen

@CHITEST führt einen Chi-Quadrat-Test durch, @FTEST führt einen F-Test durch, und @TTEST führt einen t-Test durch.

@DATUMFOLGE

@DATUMFOLGE(*Datum*) wandelt eine Datumseriennummer in das entsprechende Datum um und zeigt sie als Label im vorgegebenen Internationalen Datumsformat an.

Argumente

Datum ist eine Datumseriennummer.

Hinweise

Sie können die Vorgabeeinstellung für das internationale Datumsformat über die Ländereinstellung in der Systemsteuerung ändern.

Beispiele

Wenn das vorgegebene Internationale Datumsformat tt/mm/jj ist, ergibt **@DATUMFOLGE**(34635) das Label 28/10/94.

Ähnliche @Funktionen

@DATUMWERT berechnet die Datumseriennummer für ein Datum, das als Label eingegeben wurde. **@DATUM** berechnet die Datumseriennummer für ein bestimmtes Datum.

@LAUFZEIT, @MLAUFZEIT

@LAUFZEIT(*Valutierungstermin;Fälligkeit;Nominalzins;Rendite;[Zinstermine];[Zinstage]*) berechnet die jährliche Laufzeit einer Anleihe mit periodischen Zinszahlungen.

@MLAUFZEIT(*Valutierungstermin;Fälligkeit;Nominalzins;Rendite;[Zinstermine];[Zinstage]*) berechnet die modifizierte jährliche Laufzeit einer Anleihe nach Macauley mit periodischen Zinszahlungen.

Argumente

Valutierungstermin ist der Valutierungstermin für die Anleihe. *Valutierungstermin* ist eine Datumseriennummer.

Fälligkeit ist das Fälligkeitsdatum der Anleihe. *Fälligkeit* ist eine Datumseriennummer. Wenn *Fälligkeit* kleiner als oder gleich *Valutierungstermin* ist, ergeben **@LAUFZEIT** und **@MLAUFZEIT** den Wert **FEHLER**.

Nominalzins ist der jährliche Nominalzins für die Anleihe. *Nominalzins* ist ein beliebiger positiver Wert oder 0.

Rendite ist die jährliche Rendite. *Rendite* ist ein beliebiger positiver Wert oder 0.

Zinstermine ist ein wahlweises Argument, mit dem die Anzahl von Zinszahlungen pro Jahr angegeben wird.

Zinstermine ist ein Wert aus der folgenden Tabelle:

| <u>Zinstermine</u> | <u>Häufigkeit der Zinszahlungen</u> |
|--------------------|--|
| 1 | Jährlich |
| 2 | Halbjährlich; Vorgabe, wenn Sie das Argument weglassen |
| 4 | Vierteljährlich |
| 12 | Monatlich |

Zinstage ist ein wahlweises Argument, mit dem die Berechnungsbasis angegeben wird. *Zinstage* ist ein Wert aus der folgenden Tabelle:

| <u>Zinstage</u> | <u>Berechnungsbasis</u> |
|-----------------|--|
| 0 | 30/360; Vorgabe, wenn Sie das Argument weglassen |
| 1 | Anzahl/Anzahl |
| 2 | Anzahl/360 |
| 3 | Anzahl/365 |
| 4 | Europäisch 30/360 |

Sie können ein wahlweises Argument nicht ohne die davorstehenden Argumente benutzen.

Hinweise

Laufzeit ist die gewichtete durchschnittliche Laufzeit bis zur Fälligkeit von Cash-flows einer Anleihe. Die Gewichtungen sind der aktuelle Wert jedes Cash-flows als Bruchteil des aktuellen Wertes aller Cash-flows.

Beispiele

Eine Anleihe hat einen Valutierungstermin vom 1. Juli 1993 und ein Fälligkeitsdatum vom 1. Dezember 1998. Der halbjährliche Nominalzins beläuft sich auf 5,50% und die jährliche Rendite auf 5,61%. Die Anleihe wird auf einer Basis von 30/360 Tagen berechnet.

Um die jährliche Laufzeit der Anleihe zu berechnen, verwenden Sie folgende Formel:

@LAUFZEIT(**@DATUM**(93;7;1);**@DATUM**(98;12;1);0,055;0,0561;2;0) = 4,734591

Um die modifizierte jährliche Laufzeit der Anleihe nach Macauley zu bestimmen, verwenden Sie folgende Formel:

@MLAUFZEIT(**@DATUM**(93;7;1);**@DATUM**(98;12;1);0,055;0,0561;2;0) = 4,605410

Ähnliche @Funktionen

@STÜCKZINS berechnet den Stückzins für Anleihen, für die periodische Zinsen bezahlt werden. **@KURS** berechnet den Kurs pro 100 DM Nennwert für Anleihen, für die periodische Zinsen bezahlt werden. **@RENDITE** berechnet die Rendite für Anleihen, für die periodische Zinsen bezahlt werden.

@ISTLEER

@ISTLEER(*Ort*) testet *Ort* auf eine leere Zelle. Wenn *Ort* eine leere Zelle ist, ergibt @ISTLEER den Wert 1 (wahr); wenn *Ort* keine leere Zelle ist, ergibt @ISTLEER den Wert 0 (falsch).

Argumente

Ort ist der Name oder die Adresse einer einzelnen Zelle. Wenn Sie einen Bereich für *Ort* angeben, ergibt @ISTLEER den Wert 0 (falsch).

Beispiele

@ISTLEER(A1) = 1, wenn Zelle A1 eine leere Zelle ist

@ISTLEER(A1) = 0, wenn Zelle A1 den Wert 1,963 enthält

@ISTLEER(A1) = 0, wenn Zelle A1 das Label Ertrag enthält

@ISTLEER(A1) = 0, wenn Zelle A1 ein Justierungszeichen enthält

@MAXVERWEIS, @MINVERWEIS

@MAXVERWEIS(*Argument-Liste*) ergibt eine absolute Referenz, einschließlich des Namens der Arbeitsmappe, auf die Zelle, die den größten Wert in einer Liste mit Bereichen enthält.

@MINVERWEIS(*Argument-Liste*) ergibt eine absolute Referenz, einschließlich Dateinamen, auf die Zelle, die den kleinsten Wert in einer Liste von Bereichen enthält.

Argumente

Argument-Liste kann eine beliebige Kombination aus Bereichen sein. Trennen Sie die Bereichsnamen oder Adressen in *Argument-Liste* durch Argumenttrennzeichen.

Wenn Sie einen einzelligen Bereich in *Argument-Liste* aufnehmen möchten, muß dieser so eingegeben werden, daß er wie eine Bereichsadresse aussieht. Benutzen Sie z.B. nicht A1, sondern A1..A1.

1-2-3 ignoriert Labels und leere Zellen in *Argument-Liste*.

Wenn keine der Zellen in *Argument-Liste* Werte enthält, ergeben **@MAXVERWEIS** und **@MINVERWEIS** den Wert NV.

Beispiele

Angenommen, Ihr 1-2-3 Verzeichnis enthält die Arbeitsmappen ANGEB1.WK4, ANGEB2.WK4 und ANGEB3.WK4. Jede Arbeitsmappe enthält Angebote von einem anderen Lieferanten. Alle Arbeitsmappen wurden mit derselben Schablone erstellt, so daß die Angebotssumme in jeder Arbeitsmappe in einer Zelle namens GESAMT steht.

Mit der folgenden Formel wird ermittelt, in welcher Adresse (einschließlich Name der Arbeitsmappe) das höchste Angebot steht:

@MAXVERWEIS(<<ANGEB1.WK4>>GESAMT;<<ANGEB2.WK4>>GESAMT;<<ANGEB3.WK4>>GESAMT)

Die folgende Formel ermittelt die Adresse (einschließlich Name der Arbeitsmappe) des niedrigsten Angebots:

@MINVERWEIS(<<ANGEB1.WK4>>GESAMT;<<ANGEB2.WK4>>GESAMT;<<ANGEB3.WK4>>GESAMT)

Ähnliche @Funktionen

@HVERWEIS und **@VVERWEIS** suchen Einträge in horizontalen oder vertikalen Verweistabellen. **@INDEX** sucht den Inhalt einer Zelle, während Sie Offset-Zahlen sowohl für die Spalte als auch die Zeile angeben. **@WAHL** ersetzt eine Verweistabelle, für die nur eine Zeile erforderlich ist. **@VERGLEICH** findet die relative Position einer Zelle, die einen bestimmten Inhalt hat. **@XINDEX** findet den Inhalt einer Zelle, die mit Spalten-, Zeilen- und Arbeitsblatt-Titeln versehen ist.

@NETTOARBEITSTAGE

@NETTOARBEITSTAGE(*Anfangsdatum*;*Enddatum*;*[Freie_Tage]*;*[Wochenenden]*) berechnet die Anzahl von Tagen von *Anfangsdatum* bis *Enddatum*, ausgenommen *Wochenenden* und *Feiertage*.

Argumente

Anfangsdatum und *Enddatum* sind Datumseriennummern.

Freie_Tage ist ein wahlweises Argument, mit dem die freien Tage angegeben werden, die aus der @NETTOARBEITSTAGE-Berechnung ausgenommen werden müssen. *Freie_Tage* ist der Name oder die Adresse eines Bereichs, der Datumseriennummern enthält.

Wochenenden ist ein wahlweises Argument, mit dem angegeben wird, welche Wochentage als *Wochenende* betrachtet werden. *Wochenenden* ist eine Zeichenfolge, die die Tage, die Sie als *Wochenendtage* angeben, mit den Ganzzahlen 0 (Montag) bis 6 (Sonntag) darstellt.

So gibt z.B. "45" an, daß Freitag und Samstag *Wochenendtage* sind. Wenn Sie *Wochenenden* weglassen, benutzt 1-2-3 "56", d.h. Samstag und Sonntag werden als *Wochenendtage* betrachtet. Wenn Sie keine *Wochenenden* angeben möchten, verwenden Sie 7.

Sie können kein wahlweises Argument ohne die davorstehenden wahlweisen Argumente verwenden.

Hinweise

@NETTOARBEITSTAGE umfaßt im Ergebnis sowohl *Anfangsdatum* als auch *Enddatum*.

Wenn Sie *Wochenenden* benutzen möchten, jedoch keine *Freien_Tage*, geben Sie für *Freie_Tage* eine leere Zelle an.

Beispiele

Sie möchten feststellen, wieviele Arbeitstage zwischen Montag, dem 31. Oktober 1994 und Mittwoch, dem 30. November 1994 liegen. Der 1. November und der 17. November sollen als *Feiertage* angegeben werden, so daß Sie für diese Datumsangaben Datumseriennummern in einen Bereich namens *FREIE_TAGE* eingeben. Samstag und Sonntag sollen als *Wochenendtage* angegeben werden, so daß Sie das Argument *Wochenenden* weglassen.

@NETTOARBEITSTAGE(@DATUM(94;10;31);@DATUM(94;11;30);FREIE_TAGE) = 21

Ähnliche @Funktionen

@DAYS360 und @D360 berechnen die Anzahl von Tagen zwischen zwei Datumseriennummern. @TAGE berechnet die Anzahl von Tagen zwischen zwei Datumsangaben, wobei eine bestimmte Basis für das Zählen der Tage benutzt wird. @ARBEITSTAG berechnet das Datum, das eine bestimmte Anzahl von Tagen vor oder nach einem bestimmten Datum liegt, *Wochenenden* und *Feiertage* ausgenommen. @NÄCHSTMONAT berechnet das Datum, das eine bestimmte Anzahl von Monaten vor oder nach einem angegebenen Datum liegt.

@NÄCHSTMONAT

@NÄCHSTMONAT(*Anfangsdatum*; *Monate*; [*Tag-des-Monats*]; [*Basis*]) berechnet die Datumseriennummer für das Datum, das eine bestimmte Anzahl von *Monaten* vor oder nach *Anfangsdatum* liegt.

Argumente

Anfangsdatum ist eine Datumseriennummer.

Monate ist eine Ganzzahl. Benutzen Sie eine positive Ganzzahl, um eine Anzahl von Monaten nach *Anfangsdatum* anzugeben, bzw. eine negative Ganzzahl, um eine Anzahl von Monaten vor *Anfangsdatum* anzugeben.

Tag-des-Monats ist ein wahlweises Argument, mit dem angegeben wird, auf welchen Tag des Monats das Ergebnis von @NÄCHSTMONAT fallen soll. *Tag-des-Monats* ist ein Wert aus der folgenden Tabelle:

| <u>Tag-des-Monats</u> | <u>@NÄCHSTMONAT ergibt</u> |
|-----------------------|---|
| 0 | Ein Datum, das auf denselben Tag des Monats fällt wie <i>Anfangsdatum</i> . Wenn <i>Anfangsdatum</i> auf einen Tag des Monats fällt, der im nächsten Monat nicht vorhanden ist (wenn z.B. <i>Anfangsdatum</i> der 30. Januar 1994 und der nächste Monat der Monat Februar ist, der 28 Tage hat), gibt @NÄCHSTMONAT das Datum zurück, das auf den letzten Tag des Monats fällt. Vorgabe, wenn Sie kein Argument angeben. |
| 1 | Ein Datum, das auf den ersten Tag des Monats fällt. |
| 2 | Ein Datum, das auf den letzten Tag des Monats fällt. |

Basis ist ein wahlweises Argument, mit dem angegeben wird, welche Basis für das Zählen der Tage benutzt werden soll. *Basis* ist ein Wert aus der folgenden Tabelle:

| <u>Basis</u> | <u>Basis für das Zählen der Tage</u> |
|--------------|---|
| 0 | 30/360 |
| 1 | Anzahl/Anzahl; Vorgabe, wenn Sie das Argument weglassen |
| 2 | Anzahl/360 |
| 3 | Anzahl/365 |
| 4 | Europäisch 30/360 |

Sie können ein wahlweises Argument nicht ohne die davorstehenden wahlweisen Argumente benutzen.

Beispiele

Sie möchten das Datum ermitteln, das einen Monat nach Donnerstag, dem 7. April 1994 auf den letzten Tag des Monats fällt.

@NÄCHSTMONAT(@DATUM(94;4;7);1;2) = 34485 oder Dienstag, der 31. Mai 1994

Ähnliche @Funktionen

@DAYS360 und @D360 berechnen die Anzahl von Tagen zwischen zwei Datumseriennummern. @ARBEITSTAG berechnet ein Datum, das eine bestimmte Anzahl von Tagen vor oder nach einem bestimmten Datum liegt, Wochenenden und Feiertage ausgenommen. @NETTOARBEITSTAGE berechnet die Anzahl von Tagen zwischen zwei Datumsangaben, ausgenommen Wochenenden und Feiertage.

@SETZEFOLGE

@SETZEFOLGE(*Zeichenfolge*; *Länge*; [*Ausrichtung*]) ergibt ein Label, das eine Länge von *Länge* Zeichen hat. Das Label besteht aus einer *Zeichenfolge* und ausreichend Leerstellen, um *Zeichenfolge* wie mit *Ausrichtung* angegeben auszurichten.

Argumente

Zeichenfolge kann eine beliebige Zeichenfolge sein.

Länge kann eine Ganzzahl von 1 bis 512 sein. Wenn *Länge* kleiner ist als die Anzahl von Zeichen in *Zeichenfolge*, ergibt @SETZEFOLGE *Zeichenfolge*.

Ausrichtung ist ein wahlweises Argument, mit dem angegeben wird, wie Text ausgerichtet wird. *Ausrichtung* ist ein Wert aus der folgenden Tabelle:

| <u>Ausrichtung</u> | <u>1-2-3 richtet Zeichenfolge wie folgt aus</u> |
|--------------------|---|
| 0 | Links neben zusätzlichen Leerstellen in <i>Länge</i> ; Vorgabe, wenn Sie das Argument weglassen. |
| 1 | In der Mitte der zusätzlichen Leerzeichen in <i>Länge</i> . Wenn eine ungerade Anzahl von zusätzlichen Leerstellen vorhanden ist, fügt 1-2-3 die eine zusätzliche Leerstelle links neben <i>Zeichenfolge</i> ein. |
| 2 | Rechts neben den zusätzlichen Leerstellen in <i>Länge</i> . |

Die meisten mit Windows benutzten Schriftarten sind Proportionalchriften. Leerstellen belegen im allgemeinen weniger Platz als Buchstaben in Proportionalchriften.

Beispiele

In den folgenden Beispielen stellt jedes • eine Leerstelle dar.

@SETZEFOLGE("ABC GmbH";14) = ABC GmbH•••••

@SETZEFOLGE("ABC GmbH";14;1) = •••ABC GmbH•••

@SETZEFOLGE("ABC GmbH";14;2) = •••••ABC GmbH

Ähnliche @Funktionen

@KOMPR löscht führende, abschließende und fortlaufende Leerzeichen aus Text.

@SUMMENEGATIV

@SUMMEPOSITIV

@SUMMENEGATIV(*Argument-Liste*) addiert nur die negativen Werte in *Argument-Liste*.

@SUMMEPOSITIV(*Argument-Liste*) addiert nur die positiven Werte in *Argument-Liste*.

Argumente

Argument-Liste kann folgende Angaben in einer beliebigen Kombination enthalten: Zahlen, numerische Formeln sowie die Adressen oder Namen von Bereichen, die Zahlen oder numerische Formeln enthalten. Trennen Sie die einzelnen Elemente von *Argument-Liste* mit Argumenttrennzeichen.

Beispiele

@SUMMENEGATIV(-2;21;5;12;-2;-7) = -11

@SUMMEPOSITIV(-2;21;5;12;-2;-7) = 38

Ähnliche @Funktionen

@SUMME addiert eine Liste mit Werten. @DSUMME addiert Werte in einer Datenbanktabelle, die bestimmten Kriterien entsprechen. @ZWISCHENSUMME addiert die Werte in einer Liste und sagt @ENDSUMME, welche Werte addiert werden müssen.

@STÜCKZINS2

@STÜCKZINS2(*Valutierungstermin*; *Fälligkeit*; *Coupon*; [*Nennwert*]; [*Zinstermine*]; [*Emission*]; [*Erst-Zins*]; [*Typ*]) berechnet den Stückzins für Wertpapiere mit periodischen Zinszahlungen auf der Grundlage japanischer Konventionen.

Argumente

Valutierungstermin ist der Abrechnungstermin für das Wertpapier und wird als Datumseriennummer angegeben.

Fälligkeit ist das Fälligkeitsdatum des Wertpapiers und wird als Datumseriennummer angegeben. Wenn *Fälligkeit* kleiner als oder gleich *Valutierungstermin*, liefert @STÜCKZINS2 einen FEHLER.

Coupon ist der jährliche Zins des festverzinslichen Wertpapiers. *Coupon* ist ein beliebiger positiver Wert oder Null, geschrieben im Dezimalformat.

Nennwert ist ein wahlfreies Argument, das den Nennwert des Wertpapiers angibt, d.h. das bei Fälligkeit zu zahlende Kapital. *Nennwert* ist ein positiver Wert. Wenn Sie das Argument *Nennwert* weglassen, wird der Wert 100 verwendet.

Zinstermine ist ein wahlfreies Argument, das die Zahl der Zinszahlungen pro Jahr angibt. *Zinstermine* ist ein Wert der folgenden Tabelle:

| <u>Zinstermine</u> | <u>Zinszahlung</u> |
|--------------------|------------------------|
| 1 | Jährlich |
| 2 | Halbjährlich (Vorgabe) |

Emission ist ein wahlfreies Argument, das das Ausgabedatum des Wertpapiers als Datumseriennummer angibt. Wenn *Emission* größer als *Valutierungstermin* ist, liefert @STÜCKZINS2 das Ergebnis FEHLER.

Erst-Zins ist ein wahlfreies Argument, das den ersten Zinstermin des Wertpapiers als Datumseriennummer angibt. Wenn *Erst-Zins* kleiner oder gleich *Emission* oder größer als *Fälligkeit* ist, liefert @STÜCKZINS2 das Ergebnis FEHLER.

Typ ist ein wahlfreies Argument, das die Art der Anleihe angibt. *Typ* ist ein Wert der folgenden Tabelle:

| <u>Typ</u> | <u>Beschreibung</u> |
|------------|--|
| 0 | Japanische Staatsanleihe; Vorgabe, wenn Sie das Argument weglassen |
| 1 | Öffentliche Anleihe oder Industrieschuldverschreibung |

Sie können kein wahlfreies Argument verwenden, ohne die ihm vorausgehenden zu verwenden.

Beispiele

Sie haben am 14. Juni 1996 eine japanische Staatsanleihe für 100000 Yen gekauft. Das Emissionsdatum der Anleihe ist der 1. Mai 1990, und das Fälligkeitsdatum ist der 20. Juni 2000. Die Zinssatz beträgt 9%. Der Zins wird halbjährlich gezahlt. Das erste Zinsdatum war der 21. Dezember 1991.

Der Stückzins berechnet sich folgendermaßen:

@STÜCKZINS2(@DATUM(96;6;14),@DATUM(2000;6;20),0,09;100000;2;@DATUM(90;5;1),@DATUM(90;12;21)) = ¥4.339

Ähnliche @Funktionen

@STÜCKZINS berechnet die Stückzinsrate für Wertpapiere mit periodischen Zinszahlungen. @KURS2 berechnet den Kurs pro 100 Yen Nennwert für Wertpapiere mit periodischen Zinszahlungen auf der Grundlage japanischer Konventionen. @RENDITE2 berechnet die Rendite für Wertpapiere mit periodischen Zinszahlungen auf der Grundlage japanischer Konventionen.

@KURS berechnet den Kurs pro 100 Dollar Nennwert für eine Anleihe. @RENDITE berechnet die Rendite für Wertpapiere mit periodischen Zinszahlungen. @LAUFZEIT berechnet die jährliche Laufzeit und @MLAUFZEIT die modifizierte jährliche Laufzeit für Wertpapiere mit periodischen Zinszahlungen.

@DATALINK

@DATALINK(*Anwendungsname*; *Themennamen*; *Elementname*; [*Format*]; [*Max-Zeilen*]; [*Max-Spalten*]; [*Max-Blätter*])
erstellt eine OLE-Verknüpfung zu Daten.

Sie können die Verknüpfung ändern, indem Sie die @DATALINK-Argumente ändern.

Argumente

Anwendungsname ist der Name der Windows Anwendung, die OLE-Verknüpfungen unterstützt. Dieses Argument wird von 1-2-3 97 nicht verwendet. Es wurde bereitgestellt, um die Kompatibilität zu vorherigen Versionen von 1-2-3 zu gewährleisten.

Themennamen ist der Name der Anwendungsdatei, zu der eine Verknüpfung hergestellt werden soll.

Elementname ist der Name des Objekts in der Serveranwendung, zu der eine Verknüpfung hergestellt werden soll. Es handelt sich dabei um das Objekt in der Datei der Serveranwendung, aus dem Sie Daten über die Verknüpfung transferieren wollen.

Format bezeichnet eines der Zwischenablageformate. Mögliche *Formate* sind "Text", "WK1" oder "WK3".

Wenn Sie das Argument weglassen, verwendet 1-2-3 das Zwischenablageformat "Text".

Max-Zeilen, *Max-Spalten* und *Max-Blätter* gibt die maximale Anzahl der Zeilen, Spalten und Arbeitsblätter für den Zielbereich an. Wenn Sie *Max-Zeilen*, *Max-Spalten* oder *Max-Blätter* auslassen, verwendet 1-2-3 so viele Zeilen, Spalten und Arbeitsblätter, wie für den Zielbereich erforderlich.

Hinweise

Wenn Sie mit "Bearbeiten - Selektiv einfügen" eine Verknüpfung erzeugen und "Text, WK1, oder WK3" als Format verwenden, wird @DATALINK in der ersten Zelle des Zielbereichs der Verknüpfung angezeigt.

Verknüpfungen, die Sie erstellen, werden automatisch "DataLink 1", "DataLink 2" usw. genannt.

Weitere Informationen finden Sie unter [OLE-Verknüpfungen in 1-2-3 erstellen](#) .

Beispiele

Die folgende @DATALINK-Formel erstellt eine OLE-Verknüpfung mit der Word Pro Datei DARLEHEN.LWP.

@DATALINK("WordPro";"C:\LOTUS\WORK\WORDPRO\DARLEHEN.LWP";"!Link_Lesezeichen1")

@DATUMKONV

@DATUMKONV(*Datum;Eingabetyp;Ausgabetyt*) konvertiert Hijri- (arabisch), Farsi- (iranisch) oder hebräische Datumsangaben in das gregorianische Datumsformat bzw. umgekehrt.

Hinweise

@DATUMKONV ist für bidirektionale Sprachen spezifisch und funktioniert nur bei Computern, die bidirektionale Sprachversionen von 1-2-3 unterstützen (z.B. Hebräisch und Farsi).

@DEZIL

@DEZIL(*Teil*; *Bereich*) gibt ein gegebenes Dezil zurück.

Argumente

Teil ist eine ganze Zahl von 0 bis 10 einschließlich.

| <u>Teil</u> | <u>Gibt zurück</u> | <u>Entspricht...</u> |
|-------------|--------------------|---|
| 0 | Minimum | @PERCENTIL(0,0; <i>Bereich</i>) oder @MIN(<i>Bereich</i>) |
| 1 | Erstes Dezil | @PERCENTIL(0,1; <i>Bereich</i>) |
| 2 | Zweites Dezil | @PERCENTIL(0,2; <i>Bereich</i>) |
| 3 | Drittes Dezil | @PERCENTIL(0,3; <i>Bereich</i>) |
| 4 | Viertes Dezil | @PERCENTIL(0,4; <i>Bereich</i>) |
| 5 | Median | @PERCENTIL(0,5; <i>Bereich</i>) oder @MEDIAN(<i>Bereich</i>) |
| 6 | Sechstes Dezil | @PERCENTIL(0,6; <i>Bereich</i>) |
| 7 | Siebtes Dezil | @PERCENTIL(0,7; <i>Bereich</i>) |
| 8 | Achtes Dezil | @PERCENTIL(0,8; <i>Bereich</i>) |
| 9 | Neuntes Dezil | @PERCENTIL(0,9; <i>Bereich</i>) |
| 10 | Maximum | @PERCENTIL(1,0; <i>Bereich</i>) oder @MAX(<i>Bereich</i>) |

Bereich ist die Adresse bzw. der Name eines Bereichs, der die Werte enthält.

Hinweise

@DEZIL berechnet leere Zellen und Zellen mit Labeln als 0.

Beispiele

Ein Bereich mit dem Namen "DATEN" enthält die folgenden Daten:

34
12
60
128
67
350
206

@DEZIL(0;DATEN) = 12,0

@DEZIL(1;DATEN) = 25,2

@DEZIL(2;DATEN) = 39,2

Ähnliche @Funktionen

@PERCENTIL berechnet den x-ten Wert einer Stichprobe der Werte in einem *Bereich*. @QUARTIL gibt ein gegebenes Quartil zurück.

@MIN findet den kleinsten Wert in einer Liste. @MAX findet den größten Wert in einer Liste. @MEDIAN ermittelt den Medianwert in einer Argumentliste.

@EZIFFERNFOLGE

@EZIFFERNFOLGE(*Ziffernfolge*) konvertiert eine *digitale Zeichenfolge* von Thai-numerischen Zeichen in eine arabische numerische Zeichenfolge.

Argumente

Ziffernfolge ist eine Zeichenfolge, die Thai-numerische Zeichen enthält.

Hinweise

Verwenden Sie **@EZIFFERNFOLGE** in Kombination mit **@WERT**, um Daten, die als Thai-numerische Zeichenfolge eingegeben wurden, in numerische Werte für Berechnungen zu konvertieren.

Beispiele

@EZIFFERNFOLGE("๑๖ สิงหาคม ๓๔") = 16 สิงหาคม 34

Ähnliche @Funktionen

@THAIZIFFERNFOLGE konvertiert *Ziffernfolgen* von arabischen Zahlen in eine Thai-numerische Zeichenfolge.

@WERT setzt eine als Zeichenfolge eingegebene Zahl in den entsprechenden Wert um.

@FINDENB

@FINDENB(*Suchfolge*; *Zeichenfolge*; *Startnummer*) berechnet die Position (in Byte) in *Zeichenfolge*, an der 1-2-3 das erste Vorkommen einer *Suchfolge* findet, beginnend mit der Position (in Byte), die durch die *Startnummer* angegeben wird.

Argumente

Suchfolge und *Zeichenfolge* stehen für durch Anführungsstriche (" ") markierten Text, eine Formel, die eine Zeichenfolge zurückgibt, die Adresse oder den Namen einer Zelle, die ein Label enthält, oder eine Formel, die ein Label als Ergebnis liefert.

Startnummer ist eine Spaltennummer in Byte.

Hinweise

Wenn 1-2-3 eine *Suchfolge* in *Zeichenfolge* nicht finden kann, gibt **@FINDENB** das Ergebnis **FEHLER** zurück. **@FINDENB** gibt auch einen **FEHLER** zurück, wenn die *Startnummer* größer als die Zahl der Byte in *Zeichenfolge* oder die *Startnummer* negativ ist.

@FINDENB berücksichtigt Groß- und Kleinschreibung und Akzente. So wird beispielsweise *die Suchfolge* "e" in *Zeichenfolge* "CAMBRIDGE" nicht gefunden.

@FINDENB ist auch in der Kombination mit @MITTEB oder @ERSETZENB nützlich, um Text zu finden und zu extrahieren oder zu ersetzen.

Beispiele

@FINDENB("k";"Verbindlichkeiten";0) = 11, da sich die *Suchfolge* k an Position 11 in *Zeichenfolge* "Verbindlichkeiten" befindet.

Ähnliche @Funktionen

@SUCHEN ermittelt die Position, gezählt in Zeichen.

@PROGNOSE

@PROGNOSE(*X*; *Y-Bereich*; *X-Bereich*) ermittelt einen Prognosewert für *X* basierend auf dem linearen Trend zwischen Werten in *Y-Bereich* und *X-Bereich*.

Argumente

X kann ein beliebiger Wert sein und steht für den zu prognostizierenden Wert.

Y-Bereich ist die Adresse bzw. der Name eines Bereichs und steht für die abhängigen Werte. Wenn *Y-Bereich* nicht-numerische Daten enthält, liefert @PROGNOSE einen FEHLER.

X-Bereich ist die Adresse bzw. der Name eines Bereichs und steht für die unabhängigen Werte. Wenn *X-Bereich* nicht-numerische Daten enthält, liefert @PROGNOSE einen FEHLER.

Y-Bereich und *X-Bereich* müssen gleich groß sein, d.h. sie müssen die gleiche Zellenanzahl enthalten. Zellen in zwei Bereichen werden anhand ihrer Reihenfolge im Bereich zu Paaren zusammengefaßt. Bereiche werden von oben nach unten (von Zeile zu Zeile), dann von links nach rechts (von Spalte zu Spalte) und anschließend von Arbeitsblatt zu Arbeitsblatt geordnet. Wenn *Y-Bereich* und *X-Bereich* nicht gleich groß sind, liefert @PROGNOSE einen FEHLER.

Hinweise

Diese Berechnungen werden auch mit @REGRESSION durchgeführt.

Beispiele

Beispielwerte:

| YBEREICH | XBEREICH |
|----------|----------|
| 250 | 3 |
| 545 | 5 |
| 550 | 5 |
| 450 | 6 |
| 605 | 6 |
| 615 | 7 |

@PROGNOSE(-1;YBEREICH;XBEREICH) = -26,786

@PROGNOSE(-0;YBEREICH;XBEREICH) = 56,786

Ähnliche @Funktionen

@REGRESSION führt eine lineare Mehrfachregression durch und erzeugt die gewünschte Statistik.

@FULLP

`@FULLP(Label)` konvertiert Ein-Byte-Zeichen (ASCII-Zeichen) in *Label* in die entsprechenden japanischen Zwei-Byte-Zeichen.

Argumente

Label ist eine Zeichenfolge, die die zu konvertierenden ASCII-Zeichen enthält. Wenn *Label* keine Zeichenfolge ist, liefert `@FULLP` einen FEHLER.

Hinweise

`@FULLP` konvertiert die ASCII-Zeichen in *Label* in die entsprechenden Zwei-Byte-Zeichen. Zwei-Byte-Zeichen werden nicht konvertiert, und `@FULLP` gibt sie unverändert zurück. `@FULLP` konvertiert keine Zeichen des Katakana Ein-Byte-Zeichensatzes in Zeichen des Katakana Zwei-Byte-Zeichensatzes.

Ähnliche @Funktionen

`@HALFP` konvertiert japanische Zwei-Byte-Zeichen in Ein-Byte-Zeichen.

@ZUKWERT2

@ZUKWERT2(*Zahlungen*; *Zinssatz*; *Laufzeit*) berechnet den zukünftigen Wert einer Kapitalanlage, basierend auf einer Reihe von gleichen *Zahlungen* und einem gleichbleibenden *Zinssatz* über die Anzahl der Zahlungsperioden in der *Laufzeit* und bei vorschüssiger Rente.

Argumente

Zahlungen steht für den Betrag, der in jeder Periode gezahlt wird, und kann jeder beliebige Wert sein.

Laufzeit steht für die Zahl der Zahlungszeiträume. *Laufzeit* kann eine beliebige positive ganze Zahl sein. Dezimalbrüche werden abgeschnitten.

Zinssatz ist der periodische Zinssatz. Es kann sich dabei um einen Dezimal- oder einen Prozentwert handeln, der größer als -1 ist.

Hinweise

Der Zeitraum, der für die Berechnung von *Zinssatz* verwendet wird, muß der gleiche sein, der auch für *Laufzeit* verwendet wird. Wenn Sie z.B. die monatlichen Zahlungen berechnen wollen, geben Sie den Zinssatz und die Laufzeit in Monaten ein. Normalerweise heißt dies, daß Sie den Zinssatz durch 12 teilen und die Anzahl der Jahre der *Laufzeit* mit 12 multiplizieren müssen.

Beispiele

Zur Altersvorsorge planen Sie, für die nächsten 20 Jahre jedes Jahr 500000 Yen auf ein Konto einzuzahlen. Sie bekommen 7,5% Zinsen, die jährlich am letzten Tag des Jahres gezahlt werden. Nun möchten Sie ausrechnen, wieviel Geld Sie nach 20 Jahren auf dem Konto haben. Sie zahlen das Geld jeweils am ersten Tag des Jahres ein.

@ZUKWERT2(500000;0,075;20) = ¥23.276.266 (der Wert Ihres Kontos nach dem Ablauf von 20 Jahren)

Das gleiche Ergebnis würde auch die Formel **@ZWERT**(500000;0,075;20;1) liefern.

Ähnliche @Funktionen

@ZUKWERT berechnet den zukünftigen Wert einer Kapitalanlage, basierend auf einer Reihe von gleichen Zahlungen und einem gleichbleibenden Zinssatz über die Anzahl der Zahlungsperioden in der Laufzeit bei nachschüssiger Rente. **@ZWERT** berechnet den zukünftigen Wert einer Kapitalanlage mit einem angegebenen aktuellen Wert bei vorschüssiger oder nachschüssiger Rente.

@AKTWERT und **@AWERT** bestimmen den aktuellen Wert einer Kapitalanlage. **@NETAKTWERT** berechnet den aktuellen Nettowert einer Kapitalanlage, indem es den aktuellen Wert vom zukünftigen Wert diskontiert.

@ZWERTBETRAG

@ZWERTBETRAG(*Kapital*; *Zinssatz*; *Laufzeit*; [*Zinstermine*]) berechnet den zukünftigen Wert eines zu einem gegebenen Zinssatz angelegten Pauschalbetrags für eine gegebene Anzahl von Perioden.

Argumente

Kapital ist das ursprünglich angelegte Kapital und kann ein beliebiger Wert sein.

Zinssatz ist der periodische Zinssatz. Es kann sich dabei um einen Dezimal- oder einen Prozentwert handeln.

Laufzeit ist die Zahl der Perioden für die Kapitalanlage und kann ein beliebiger Wert sein.

Zinstermine ist ein wahlfreies Argument und gibt die Häufigkeit pro Periode an. Dezimalbrüche werden abgeschnitten. Die Vorgabehäufigkeit ist 1.

Sie können kein wahlfreies Argument verwenden, ohne die ihm vorausgehenden zu verwenden.

Hinweise

Wenn *Laufzeit* gleich 1 und *Zinstermine* gleich 1 ist, berechnet **@ZWERTBETRAG** die einfachen Zinsen.

Beispiel

Nehmen Sie einmal an, daß Sie am 10. Geburtstag Ihres Kindes 100 DM auf einem Konto mit einem Zinssatz von 10% anlegen. Mit **@ZWERTBETRAG** können Sie errechnen, auf welchen Betrag diese 100 DM angewachsen sind, wenn Ihr Kind 18 Jahre alt wird:

@ZWERTBETRAG(100;10%;8) = 214,36 DM

Ähnliche @Funktionen

@ZWERTBETRAG bezieht sich auf eine einmalige Zahlung. **@ZUKWERT** und **@ZWERT** berechnen den zukünftigen Wert bei Rentenzahlung, d.h. bei einer oder mehreren Zahlungen. **@AWERTBETRAG** berechnet den aktuellen Wert eines angelegten Pauschalbetrags für eine gegebene Anzahl von Perioden bei Diskontierung eines gegebenen Zinssatzes.

@HALFP

@HALFP(*Label*) konvertiert japanische Zwei-Byte-Zeichen in *Label* in die entsprechenden Ein-Byte-Zeichen (ASCII-Zeichen).

Argumente

Label ist eine Zeichenfolge, die die zu konvertierenden japanischen Zwei-Byte-Zeichen enthält. Wenn *Label* keine Zeichenfolge ist, liefert **@HALFP** einen FEHLER.

Hinweise

@HALFP konvertiert Buchstaben, Nummern und Zeichen (z.B. +,-,:) des japanischen Zwei-Byte-Zeichensatzes in die entsprechenden Ein-Byte-Zeichen (ASCII-Zeichen). ASCII-Zeichen und Zeichen, für die es keine ASCII-Entsprechung gibt, werden nicht konvertiert, sondern von **@HALFP** unverändert zurückgegeben. **@HALFP** konvertiert keine Zeichen des Katakana Zwei-Byte-Zeichensatzes in Zeichen des Katakana Ein-Byte-Zeichensatzes.

Ähnliche @Funktionen

@FULLP konvertiert Ein-Byte-Zeichen in japanische Zwei-Byte-Zeichen.

@ISTZWISCHEN

@ISTZWISCHEN(*Wert*; *Grenze1*; *Grenze2*; [*Typ*]) prüft, ob ein *Wert* zwischen *Grenze1* und *Grenze2* liegt.

@ISTZWISCHEN gibt 1 (wahr) zurück, wenn der *Wert* zwischen *Grenze1* und *Grenze2* liegt, und 0 (falsch), wenn dies nicht der Fall ist.

Argumente

Wert, *Grenze1* und *Grenze2* können jeder beliebige numerische Wert oder Text sein.

Typ ist wahlfrei und kann eines der folgenden Elemente sein:

| <u>Typ</u> | <u>Beschreibung</u> | <u>Algorithmus</u> |
|------------|---|---|
| 0 | <i>Grenze1</i> und <i>Grenze2</i> einschließen (Vorgabe) | @MIN(<i>Grenze1</i> , <i>Grenze2</i>) <= <i>Wert</i> <= @MAX(<i>Grenze1</i> , <i>Grenze2</i>) |
| 1 | Das Kleinste von <i>Grenze1</i> und <i>Grenze2</i> einschließen und das Größte ausschließen | @MIN(<i>Grenze1</i> ; <i>Grenze2</i>) <= <i>Wert</i> < @MAX(<i>Grenze1</i> , <i>Grenze2</i>) |
| 2 | Das Kleinste von <i>Grenze1</i> und <i>Grenze2</i> ausschließen und das Größte einschließen | @MIN(<i>Grenze1</i> , <i>Grenze2</i>) <= <i>Wert</i> <= @MAX(<i>Grenze1</i> , <i>Grenze2</i>) |
| 3 | Sowohl <i>Grenze1</i> als auch <i>Grenze2</i> ausschließen | @MIN(<i>Grenze1</i> , <i>Grenze2</i>) < <i>Wert</i> < @MAX(<i>Grenze1</i> , <i>Grenze2</i>) |

Hinweise

Wenn sich der Datentyp von *Grenze1* vom Datentyp von *Grenze2* unterscheidet, liefert @ISTZWISCHEN einen FEHLER.

Wenn sich der Datentyp von *Grenze1* oder *Grenze2* vom Datentyp von *Wert* unterscheidet, liefert @ISTZWISCHEN das Ergebnis 0.

Sie können kein wahlfreies Argument verwenden, ohne die ihm vorausgehenden zu verwenden.

Beispiele

@ISTZWISCHEN(123;100;200) = 1

@ISTZWISCHEN("ABC";"AAA";"BBB") = 1

@ISTZWISCHEN(100;100;200;0) = 1

@LINKSB

@LINKSB(*Zeichenfolge*; *n*) gibt die ersten *n* Bytes in *Zeichenfolge* zurück.

Argumente

Zeichenfolge ist durch Anführungsstriche (" ") markierter Text, eine Formel, die eine Zeichenfolge als Ergebnis liefert, die Adresse oder der Name einer Zelle, die ein Label enthält, oder eine Formel, die ein Label als Ergebnis liefert.

n kann eine positive ganze Zahl oder 0 sein. Wenn *n* gleich 0 ist, ist das Resultat eine leere Zeichenfolge. Wenn *n* größer als die Byte-Länge von *Zeichenfolge* ist, gibt @LINKSB die gesamte *Zeichenfolge* zurück.

Hinweise

@LINKSB zählt Satz- und Leerzeichen mit.

@LINKSB wird für Zwei-Byte-Zeichensätze (DBCS), z. B. Japanisch, verwendet. Wenn Sie z.B. aus 1-2-3 ein Label in ein Feld in einer externen Datenbank einfügen wollen und das Feld eine Längenbegrenzung in Byte hat, können Sie Teile des Labels extrahieren und so an diese Begrenzung anpassen.

Wenn es sich beim letzten Byte des zurückgegebenen Labels um die erste Hälfte eines Zwei-Byte-Zeichens handelt, ersetzt @LINKSB das unvollständige Zeichen durch ein Leerzeichen.

Wenn *Zeichenfolge* nur aus Ein-Byte-Zeichen besteht, gibt @LINKSB das gleiche Ergebnis zurück wie @LINKS.

Beispiele

@LINKSB("Ein Byte";5) = "Ein B"

Ähnliche @Funktionen

@LINKS gibt die ersten *n* Zeichen in *Zeichenfolge* zurück. @MITTEB gibt Bytes aus *Zeichenfolge* zurück.

@RECHTSB gibt die letzten *n* Bytes in *Zeichenfolge* zurück.

@LÄNGEB

@LÄNGEB (*Zeichenfolge*) zählt die Anzahl der Bytes in *Zeichenfolge*.

Argumente

Zeichenfolge ist durch Anführungsstriche (" ") markierter Text, eine Formel, die eine Zeichenfolge als Ergebnis liefert, die Adresse oder der Name einer Zelle, die ein Label enthält, oder eine Formel, die ein Label als Ergebnis liefert.

Hinweise

@LÄNGEB zählt Satz- und Leerzeichen mit.

@LÄNGEB wird für Zwei-Byte-Zeichensätze (DBCS), z. B. Japanisch, verwendet. Wenn Sie z.B. aus 1-2-3 eine Zeichenfolge in ein Feld in einer externen Datenbank einfügen wollen und das Feld eine Längenbegrenzung in Byte hat, können Sie prüfen, ob Teile der Zeichenfolge die Begrenzung überschreiten.

Beispiele

@LÄNGEB(A5&G12) = Gesamtzahl der Bytes in den Zellen A5 und G12.

Ähnliche @Funktionen

@LÄNGE zählt die Zeichen in *Zeichenfolge*.

@MITTEB

@MITTEB(*Zeichenfolge*; *Startnummer*; *n*) kopiert *n* Bytes aus *Zeichenfolge*, beginnend mit den Daten an der Position *Startnummer* (in Byte).

Argumente

Zeichenfolge ist durch Anführungsstriche (" ") markierter Text, eine Formel, die eine Zeichenfolge als Ergebnis liefert, die Adresse oder der Name einer Zelle, die ein Label enthält, oder eine Formel, die ein Label als Ergebnis liefert.

Startnummer ist eine Spaltennummer in Byte. Wenn *Startnummer* größer als die Länge von *Zeichenfolge* ist, ist das Ergebnis von @MITTEB eine leere Zeichenfolge.

n ist eine beliebige positive ganze Zahl oder 0. Wenn *n* gleich 0 ist, ist das Ergebnis von @MITTEB eine leere Zeichenfolge. Wenn *n* länger als die Länge von *Zeichenfolge* ist, gibt 1-2-3 alle Byte von der *Startnummer* bis zum Ende von *Zeichenfolge* zurück.

Hinweise

@MITTEB wird für Zwei-Byte-Zeichensätze (DBCS), z. B. Japanisch, verwendet.

@MITTEB zählt Satz- und Leerzeichen mit.

Verwenden Sie eine große Zahl für *n*, wenn Sie die Länge von *Zeichenfolge* nicht kennen; 1-2-3 ignoriert die zusätzlichen Leerzeichen und gibt die gesamte *Zeichenfolge* zurück, beginnend mit der *Startnummer*.

Wenn das erste Byte des zurückgegebenen Labels die letzte Hälfte eines Zwei-Byte-Zeichens oder das letzte Byte des zurückgegebenen Labels die erste Hälfte eines Zwei-Byte-Zeichens ist, ersetzt @MITTEB das unvollständige Zeichen durch ein Leerzeichen.

Beispiele

@MITTEB("Ein-Byte-Zeichenfolge";4;4) = "Byte"

Ähnliche @Funktionen

@MITTE kopiert *n* Zeichen aus einer Zeichenfolge, beginnend mit dem Zeichen bei *Startnummer*.

@LINKSB gibt die ersten *n* Byte von *Zeichenfolge* zurück, und @RECHTSB gibt die letzten *n* Byte von *Zeichenfolge* zurück.

@NSUMME

@NSUMME(*Versatz*; *n*; *Argumentliste*) summiert jeden *n*-ten Wert in *Argumentliste*, beginnend bei *Versatz*.

@NSUMME ignoriert FEHLER und NV in übersprungenen Zellen/Werten.

Argumente

Argumentliste kann die folgenden Elemente in beliebiger Kombination enthalten: Zahlen, numerische Formeln sowie Adressen und Namen von Bereichen, die Zahlen oder Formeln enthalten. Trennen Sie die Elemente der *Argumentliste* mit Argumenttrennzeichen voneinander ab.

Versatz kann eine beliebige positive ganze Zahl sein.

n kann eine beliebige positive ganze Zahl sein.

Hinweise

@NSUMME gibt die Summe von (*Versatz*), (*Versatz* + *n*), (*Versatz* + 2*n*), (*Versatz* + 3*n*) ... in *Argumentliste* zurück.

@NSUMME(0;1;*Argumentliste*) gibt das gleiche Ergebnis wie **@SUMME**(*Argumentliste*) zurück.

Beispiele

@NSUMME(1;3;B5..B15) ermittelt die Summe der Werte in B6, B9, B12 und B15.

Ähnliche @Funktionen

@SUMME summiert die Werte in *Argumentliste*. **@DSUMME** berechnet die Summe von Werten, die die von Ihnen festgelegten Kriterien erfüllen. **@ZWISCHENSUMME** summiert die Werte in einer Liste und zeigt an, welche Werte **@ENDSUMME** summieren soll.

@ZAHLENFOLGE

@ZAHLENFOLGE(*Zahl*; *Typ*) konvertiert *Zahl* in eine japanische numerische Zeichenfolge und verwendet dabei das in *Typ* angegebene Format.

Argumente

Zahl kann jede beliebige gerundete Zahl zwischen 0 und 9.999.999.999.999 sein. 1-2-3 rundet *Zahl* auf die nächste Ganzzahl auf bzw. ab. Wenn Sie eine Zahl angeben, die außerhalb des Bereichs liegt, liefert @ZAHLENFOLGE einen FEHLER.

Typ legt das Format fest. Verwenden Sie einen der Werte der folgenden Tabelle:

Typ Gibt zurück

- | | |
|---|---|
| 1 | Normale Kanji-Zahlenschreibweise |
| 2 | Formale Kanji-Zahlenschreibweise |
| 3 | Einfache Eins-zu-Eins-Umsetzung in Kanji-Zahlen |

Beispiele

@ZAHLENFOLGE(1234567890,1) = 十二億三千四百五十六万七千八百九十

@ZAHLENFOLGE(1234567890,2) = 拾貳億參千四百五拾六万七千八百九拾

@ZAHLENFOLGE(1234567890,3) = 一 二 三 四 五 六 七 八 九 〇

@RATE2

@RATE2(*Kapital*; *Zinssatz*; *Laufzeit*) berechnet die Zahlung für ein Darlehen (*Kapital*) bei einem gegebenen *Zinssatz* für eine festgelegte Zahl von Zahlungsperioden (*Laufzeit*) und bei vorschüssiger Rente.

Argumente

Kapital ist der Darlehensbetrag und kann ein beliebiger Wert sein.

Zinssatz ist der Zinssatz für den Kredit. Es kann sich dabei um einen Dezimal- oder einen Prozentwert handeln, der größer als -1 ist.

Laufzeit steht für die Zahl der Zahlungszeiträume. *Laufzeit* kann eine beliebige positive ganze Zahl sein. Dezimalbrüche werden abgeschnitten.

Hinweise

Der Zeitraum, der für die Berechnung von *Zinssatz* verwendet wird, muß der gleiche sein, der auch für *Laufzeit* verwendet wird. Wenn Sie z.B. die monatlichen Zahlungen berechnen wollen, geben Sie den Zinssatz und die Laufzeit in Monaten ein. Normalerweise heißt dies, daß Sie den Zinssatz durch 12 teilen und die Anzahl der Jahre der *Laufzeit* mit 12 multiplizieren müssen.

Im Gegensatz zu **@RATE** verwendet **@RATE2** für die Berechnung nicht den Beginn, sondern das Ende der Laufzeit.

Beispiele

Sie überlegen sich, ein Darlehen von 8000 Yen mit einem jährlichen Zinssatz von 10,5% und einer Laufzeit von 3 Jahren aufzunehmen. Die Tilgung erfolgt monatlich. Die Zahlungen müssen am letzten Tag eines jeden Monats geleistet werden. So berechnen Sie die monatlich zu leistenden Zahlungen:

@RATE2(8000;0,105/12;36) = ¥257,76

@ANNUITÄT(8000;0,105/12;36;1) gibt das gleiche Ergebnis zurück.

Ähnliche @Funktionen

@RATE und **@ANNUITÄT** berechnen die Zahlungen für ein Darlehen (*Kapital*) bei einem gegebenen Zinssatz für eine festgelegte Anzahl an Zahlungsperioden und bei nachschüssiger Rente. **@RATEZ** berechnet den Zinsanteil von periodischen Zahlungen.

@ZINSBETRAG berechnet den kumulativen Zinsanteil der periodischen Zahlung für eine Investition. **@TILGUNG** berechnet den Kapitalanteil der periodischen Zahlung für eine Investition. **@ANN** berechnet die Zahl der Zahlungsperioden für eine Investition. **@ANN2** berechnet die Zahl der erforderlichen Perioden für eine Reihe gleicher Zahlungen zur Akkumulation einer bestimmten Summe bei gegebenem Zinssatz und bei vorschüssiger Rente.

@RATEZ

@RATEZ(*Kapital;Zinssatz;Laufzeit;Periode*) berechnet den Zinsanteil bei einer konstanten periodischen Zahlung.

Argumente

Kapital ist der Darlehensbetrag und kann ein beliebiger Wert sein.

Zinssatz ist der Zinssatz für den Kredit. Es kann sich dabei um einen Dezimal- oder einen Prozentwert handeln, der größer als -1 ist.

Laufzeit steht für die Zahl der Zahlungszeiträume. *Laufzeit* kann eine beliebige positive ganze Zahl sein. Dezimalbrüche werden abgeschnitten.

Zeitraum ist der Zahlungszeitraum, für den Sie die Zinszahlungen bestimmen wollen. *Periode* kann jede beliebige ganze Zahl sein, die kleiner oder gleich *Laufzeit* ist. Dezimalbrüche werden abgeschnitten.

Hinweise

Der Zeitraum, der für die Berechnung von *Zinssatz* verwendet wird, muß der gleiche sein, der auch für *Laufzeit* verwendet wird. Wenn Sie z.B. die monatlichen Zahlungen berechnen wollen, geben Sie den Zinssatz und die Laufzeit in monatlichen Schritten ein. Normalerweise heißt dies, daß Sie den Zinssatz durch 12 teilen und die Anzahl der Jahre mit 12 multiplizieren müssen.

@ZINSBETRAG(*Kapital;Zinssatz;Laufzeit;Periode;Periode;0*) liefert das gleiche Ergebnis wie **@RATEZ**(*Kapital;Zinssatz;Laufzeit;Periode*).

Beispiele

Sie haben ein Darlehen von \$8000 bei einem jährlichen Zinssatz von 10,5% und einer Laufzeit von 3 Jahren aufgenommen. Die Tilgung erfolgt monatlich. Ihre monatlichen Zahlungen betragen 260,02 Dollar. Den Zinsanteil der ersten Zahlung berechnen Sie folgendermaßen:

@RATEZ(8000;0,105/12;36;1) = \$70

Ähnliche @Funktionen

@RATE und **@ANNUITÄT** berechnen die Zahlungen für ein Darlehen (*Kapital*) bei einem gegebenen Zinssatz und bei nachschüssiger Rente. **@RATE2** berechnet die Zahlungen für ein Darlehen bei gegebenem Zinssatz für eine festgelegte Anzahl von Perioden und bei vorschüssiger Rente.

@SPI berechnet den Zinsanteil von periodischen Zahlungen für ein Darlehen bei gleichbleibender Tilgungssumme je Zahlungsperiode.

@ZINSBETRAG berechnet den kumulativen Zinsanteil bei periodischen Zahlungen für ein Darlehen bei gegebenem Zinssatz für eine festgelegte Anzahl an Zahlungsperioden.

@KURS2

@KURS2(*Valutierungstermin*; *Fälligkeit*; *Coupon*; *Rendite*; [*Rückzahlung*]; [*Zinstage*]) berechnet den Kurs pro 100 Yen Nennwert für Wertpapiere mit periodischen Zinszahlungen und auf der Grundlage japanischer Konventionen.

Argumente

Valutierungstermin ist der Abrechnungstermin für das Wertpapier. *Valutierungstermin* ist eine Datumseriennummer.

Fälligkeit ist das Fälligkeitsdatum des Wertpapiers. *Fälligkeit* ist eine Datumseriennummer. Wenn *Fälligkeit* kleiner als oder gleich *Valutierungstermin* ist, liefert @KURS2 einen FEHLER.

Coupon ist der jährliche Zinssatz des Wertpapiers, angegeben als Dezimalwert. *Coupon* ist ein beliebiger positiver Wert oder 0.

Rendite ist die jährliche Rendite, angegeben als Dezimalwert. *Rendite* kann ein beliebiger positiver Wert sein.

Rückzahlung ist ein wahlfreies Argument, das den Rückzahlungswert des Wertpapiers pro 100 Yen Nennwert angibt.

Rückzahlung ist ein beliebiger positiver Wert oder 0. Wenn Sie das Argument *Rückzahlung* weglassen, wird der Wert 100 verwendet.

Zinstage ist ein wahlfreies Argument, das die Art der verwendeten Berechnungsbasis angibt:

| <u>Zinstage</u> | <u>Berechnungsbasis</u> |
|-----------------|-------------------------|
| 0 | 30/360 |
| 1 | Anzahl/Anzahl |
| 2 | Anzahl/360 |
| 3 | Anzahl/365 |
| 4 | Europäisch 30/360 |

Wenn Sie das Argument *Zinstage* verwenden, ruft 1-2-3 die Funktion @KURS auf, um den Wert zu berechnen.

Beispiele

Eine Anleihe hat den Valutierungstermin 18. Juli 1996. Das Fälligkeitsdatum ist der 20. Juni 2000. Die Zinsrate beträgt 5,50% und die jährliche Rendite 5,61%. Den Kurs der Anleihe berechnen Sie folgendermaßen:

@KURS2(@DATUM(96;7;18),@DATUM(100;6;20),0,055;0,0561;100) = ¥99,65

Ähnliche @Funktionen

@KURS berechnet den Kurs pro 100 Dollar Nennwert für Wertpapiere mit periodischen Zinszahlungen.

@STÜCKZINS berechnet die Stückzinsen für Wertpapiere mit periodischen Zinszahlungen. @RENDITE berechnet die Rendite für Wertpapiere mit periodischen Zinszahlungen. @LAUFZEIT berechnet die jährliche Laufzeit und @MLAUFZEIT die modifizierte jährliche Laufzeit für Wertpapiere mit periodischen Zinszahlungen.

@STÜCKZINS2 berechnet den Stückzins für Wertpapiere mit periodischen Zinszahlungen unter Verwendung japanischer Konventionen. @RENDITE2 berechnet die Rendite für Wertpapiere mit periodischen Zinszahlungen auf der Grundlage japanischer Konventionen.

@PRODUKT

@PRODUKT(*Argumentliste*) multipliziert die Werte in *Argumentliste*.

Argumente

Argumentliste kann eine der folgenden Angaben in einer beliebigen Kombination enthalten: Zahlen, numerische Formeln und Adressen oder Namen von Bereichen, die Zahlen oder numerische Formeln enthalten. Trennen Sie die Elemente der *Argumentliste* mit Argumenttrennzeichen.

Siehe auch Argumente für statistische @Funktionen.

Beispiele

@PRODUKT(2;4;6;8) = 384

Ähnliche @Funktionen

@FAKULTÄT berechnet den Faktor eines Wertes. @SUMME addiert die Werte in einer Argumentliste.

@SUMPROD berechnet die Summe der Produkte entsprechender Werte in mehreren Bereichen.

@NURMEDIAN

@NURMEDIAN(*Argumentliste*) gibt den Medianwert in *Argumentliste* zurück und ignoriert dabei leere Zellen, Label und Formeln, die Label zurückgeben.

Argumente

Argumentliste kann die folgenden Elemente in beliebiger Kombination enthalten: Zahlen, numerische Formeln sowie Adressen oder Namen von Bereichen, die Zahlen oder numerische Formeln enthalten. Trennen Sie Elemente von *Argumentliste* mit Argumenttrennzeichen.

Hinweise

Median ist der Mittelwert in *Argumentliste*, d.h., es gibt eine gleiche Anzahl an kleineren und an größeren Werten als der Medianwert. Wenn *Argumentliste* eine ungerade Anzahl an Werten enthält, sortiert **@NURMEDIAN** die Werte und gibt den Mittelwert zurück. Wenn *Argumentliste* ein gerade Anzahl an Werten enthält, sortiert **@NURMEDIAN** die Werte und gibt das arithmetische Mittel der beiden Mittelwerte zurück.

@NURMEDIAN liefert FEHLER, wenn der Bereich leer ist oder nur Label bzw. Formeln enthält, die ein Label zurückgeben.

Beispiele

@NURMEDIAN(A2..A6) = 12, wenn A2..A6 die Werte 5, 12, 65, 82 und 9 enthält.

@NURMEDIAN(A1..A6) = 12, wenn A1..A6 die Werte 5, 12, 65, 82 und 9 und das Label "Januar" enthält.

(**@NURMEDIAN** ignoriert das Label.) **@MEDIAN** würde das Label als 0 berechnen und somit den Wert 10,5 für diese Daten zurückgeben.

Ähnliche @Funktionen

@MEDIAN ermittelt den Medianwert in einer Argumentliste.

@AKTWERT2

@AKTWERT2(*Zahlungen;Zinssatz;Laufzeit*) berechnet den aktuellen Wert einer Investition, basierend auf einer Reihe gleicher *Zahlungen* bei Diskontierung periodischer *Zinssätze* über die Anzahl der Perioden in *Laufzeit* bei vorschüssiger Rente.

Argumente

Zahlungen steht für die Zahlungsrate und kann jeder beliebige Wert sein.

Zinssatz ist der Zinssatz der Investition. Es kann sich dabei um einen Dezimal- oder einen Prozentwert handeln, der größer als -1 ist.

Laufzeit steht für die Zahl der Zahlungszeiträume. *Laufzeit* kann eine beliebige positive ganze Zahl sein. Dezimalbrüche werden abgeschnitten.

Hinweise

Der Zeitraum, der für die Berechnung von *Zinssatz* verwendet wird, muß der gleiche sein, der auch für *Laufzeit* verwendet wird. Wenn Sie z.B. die monatlichen Zahlungen berechnen wollen, geben Sie den Zinssatz und die Laufzeit in Monaten ein. Normalerweise heißt dies, daß Sie den Zinssatz durch 12 teilen und die Anzahl der Jahre der *Laufzeit* mit 12 multiplizieren müssen.

Im Gegensatz zu **@AKTWERT** verwendet **@AKTWERT2** für die Berechnung nicht den Beginn, sondern das Ende der Laufzeit.

Verwenden Sie **@AKTWERT2**, um eine Investition zu bewerten oder mit anderen Investitionen zu vergleichen. **@AKTWERT2** ist für den Vergleich verschiedener Kapitalanlageformen nützlich. So können Sie beispielsweise eine einmalige Anlage mit einem Rentenfonds mit periodischen Zahlungen vergleichen. Verwenden Sie **@AKTWERT2** zusammen mit **@RATE2**, um eine Amortisationstabelle zu erstellen.

@AKTWERT2 ergänzt **@RATE**. **@AKTWERT2** berechnet ausgehend von der Höhe der monatlichen Zahlungen, die Sie sich leisten können, die Höhe des Darlehens, das Sie aufnehmen können. Dahingegen berechnet **@RATE**, wie hoch die monatlichen Zahlungen abhängig von der Darlehenssumme sind, die Sie aufnehmen möchten.

Beispiele

Sie haben 1.000.000 Yen gewonnen. Sie haben zwei Möglichkeiten: Sie können entweder 20 Jahre lang am Ende eines jeden Jahres 50.000 Yen beziehen oder sich anstelle der Rente in Höhe von 1.000.000 Yen eine einmalige Summe von 400.000 Yen auszahlen lassen. Sie möchten nun wissen, welche Option, gemessen am heutigen Wert des Yen, die rentablere ist.

Nehmen wir an, Sie akzeptieren die jährliche Rente in Höhe von 50.000 Yen und legen das Geld zu einem Zinssatz von 8% an. Die Zinsen werden jährlich ausgezahlt.

@AKTWERT2(50000;0,08;20) ergibt 530.180 Yen, d.h. die 1.000.000 sind, über 20 Jahre ausbezahlt, nach zum aktuellen Datum ¥530.180 wert.

Das gleiche Ergebnis würde auch die Formel **@AWERT**(50000;0,08;20;1) liefern.

Ähnliche @Funktionen

@AKTWERT berechnet den aktuellen Wert einer Kapitalanlage, basierend auf einer Reihe von gleichen Zahlungen bei Diskontierung der periodischen Zinsen über die Perioden in der Laufzeit bei nachschüssiger Rente. **@AWERT** berechnet den aktuellen Wert einer Kapitalanlage mit einem angegebenen zukünftigen Wert bei vorschüssiger oder nachschüssiger Rente.

@ZUKWERT und **@ZWERT** berechnen den zukünftigen Wert einer Kapitalanlage, basierend auf einer Reihe von gleichen Zahlungen. **@ZUKWERT2** berechnet den zukünftigen Wert einer Kapitalanlage, basierend auf einer Reihe von gleichen Zahlungen und einem gleichbleibenden Zinssatz über die Anzahl der Zahlungsperioden in der Laufzeit bei vorschüssiger Rente. **@NETAKTWERT** berechnet den aktuellen Nettowert einer Kapitalanlage, indem es den aktuellen Wert vom zukünftigen Wert diskontiert.

@RATE und **@ANNUITÄT** berechnen die Zahlungen für ein Darlehen bei einem gegebenen Zinssatz für eine festgelegte Anzahl an Zahlungsperioden. **@RATE2** berechnet die Zahlungen für ein Darlehen bei gegebenem Zinssatz für eine festgelegte Anzahl von Perioden und bei vorschüssiger Rente.

@AWERTBETRAG

@AWERTBETRAG(*Zukunft*; *Zinssatz*; *Laufzeit*; [*Zinstermine*]) berechnet den aktuellen Wert eines angelegten Pauschalbetrags für eine gegebene Anzahl von Perioden bei Diskontierung eines gegebenen Zinssatzes.

Argumente

Zukunft steht für die in der Zukunft zu erwartende Summe und kann ein beliebiger Wert sein.

Zinssatz ist der periodische Diskontsatz. Es kann sich dabei um einen Dezimal- oder einen Prozentwert handeln.

Laufzeit steht für die Anzahl der Perioden bis zur Auszahlung des Kapitals und kann ein beliebiger Wert sein.

Zinstermine ist ein wahlfreies Argument und gibt die Häufigkeit pro Periode an. Dezimalbrüche werden abgeschnitten. Die Vorgabehäufigkeit ist 1.

Hinweise

Wenn *Laufzeit* gleich 1 und *Zinstermine* gleich 1 ist, berechnet **@AWERTBETRAG** die einfachen Zinsen.

Beispiele

Nehmen Sie an, Sie benötigen 10.000 Dollar für die universitäre Ausbildung Ihres Kindes. Diese sollen in zehn Jahren ausgezahlt werden, und Sie erhalten für das Geld 10% Zinsen. Mit **@AWERTBETRAG** können Sie bestimmen, welchen Pauschalbetrag Sie heute anlegen müssen:

@AWERTBETRAG(10000;10%,10) = \$3855,43

Ähnliche @Funktionen

@AKTWERT und **@PVAL** berechnen den aktuellen Wert bei Rentenzahlung, d.h. bei einer oder mehreren Zahlungen. **@ZWERTBETRAG** berechnet den zukünftigen Wert eines bei einem bestimmten Zinssatz angelegten Pauschalbetrags für eine gegebene Anzahl von Perioden.

@QUARTIL

@QUARTIL(*Teil*; *Bereich*) gibt ein gegebenes Quartil zurück.

Argumente

Teil ist eine ganze Zahl von 0 bis 4 einschließlich. Verwenden Sie für *Teil* einen der Werte der folgenden Tabelle:

| <u>Teil</u> | <u>Gibt zurück</u> | <u>Entspricht...</u> |
|-------------|--------------------|---|
| 0 | Minimum | @PERCENTIL(0,0; <i>Bereich</i>) oder @MIN(<i>Bereich</i>) |
| 1 | Erstes Quartil | @PERCENTIL(0,25; <i>Bereich</i>) |
| 2 | Median | @PERCENTIL(0,50; <i>Bereich</i>) oder @MEDIAN(<i>Bereich</i>) |
| 3 | Drittes Quartil | @PERCENTIL(0,75; <i>Bereich</i>) |
| 4 | Maximum | @PERCENTIL(1,00; <i>Bereich</i>) oder @MAX(<i>Bereich</i>) |

Bereich ist die Adresse bzw. der Name des Bereichs, der die Werte enthält.

Hinweise

@QUARTIL berechnet leere Zellen und Zellen mit Labeln als 0.

Beispiele

"Daten" ist ein Bereich, der die folgenden Daten enthält:

34
12
60
128
67
350
206

@QUARTIL(0;DATEN) = 12

@QUARTIL(1;DATEN) = 47

@QUARTIL(2;DATEN) = 67

@QUARTIL(3;DATEN) = 167

@QUARTIL(4;DATEN) = 350

Ähnliche @Funktionen

@PERCENTIL berechnet den x-ten Wert einer Stichprobe der Werte in *Bereich*. @DEZIL gibt ein gegebenes Dezil zurück.

@ERSETZENB

@ERSETZENB(*Original-Zeichenfolge*; *Startnummer*; *n*; *Ersatz-Zeichenfolge*) ersetzt *n* Byte in *Original-Zeichenfolge* mit *Ersatz-Zeichenfolge*, beginnend mit *Startnummer*.

Argumente

Original-Zeichenfolge und *Ersatz-Zeichenfolge* stehen für durch Anführungsstriche (" ") markierten Text, Formeln, die eine Zeichenfolge zurückgeben, Adressen oder Namen von Zellen, die Label enthalten, oder Formeln, die Label zurückgeben.

Startnummer ist die Spaltennummer der Daten in *Original-Zeichenfolge*. Es kann sich dabei um einen beliebigen positiven Wert oder 0 handeln. Wenn *Startnummer* größer als die Byte-Länge von *Original-Text* ist, hängt **@ERSETZENB** die *Ersatz-Zeichenfolge* an *Original-Zeichenfolge* an.

n kann eine beliebige positive ganze Zahl oder 0 sein. Wenn *n* gleich 0 ist, fügt **@ERSETZENB** die *Ersatz-Zeichenfolge* bei *Startnummer* ein, ohne Daten in *Original-Zeichenfolge* zu löschen.

Hinweise

@ERSETZENB handhabt Positionen und Längen von Zeichenfolgen in Byte.

@ERSETZENB zählt Satz- und Leerzeichen mit. Wenn Sie **@ERSETZENB** verwenden, um Text anzuhängen oder einzufügen, müssen Sie die erforderlichen Leerzeichen einfügen.

Wenn die ersetzten Daten in *Original-Zeichenfolge* in der Mitte eines Zwei-Byte-Zeichens beginnen oder enden, ersetzt **@ERSETZENB** jedes unvollständige Zwei-Byte-Zeichen durch ein Leerzeichen im zurückgegebenen Text.

Verwenden Sie **@FINDENB** zusammen mit **@ERSETZENB**, um ein Label zu suchen und zu ersetzen oder um eine unbekannt *Startnummer* zu berechnen.

@ERSETZENB ist nützlich, wenn Sie bestimmte Zeichen durch andere ersetzen müssen, z.B. die Vorwahl in einer Datenbank mit Telefonnummern.

Beispiele

@ERSETZENB(ZELLE;**@FINDENB**("-";ZELLE;0),1;"") ersetzt das Label in ZELLE, 4-24, durch 4/24.

Ähnliche @Funktionen

@ERSETZEN ersetzt *n* Zeichen in *Original-Zeichenfolge* durch *Ersatz-Zeichenfolge*, beginnend bei *Startnummer*.

@RECHTSB

@RECHTSB(*Zeichenfolge*; *n*) gibt die letzten *n* Byte in einer *Zeichenfolge* zurück.

Argumente

Zeichenfolge ist durch Anführungsstriche (" ") markierter Text, eine Formel, die eine *Zeichenfolge* als Ergebnis liefert, die Adresse oder der Name einer Zelle, die ein Label enthält, oder eine Formel, die ein Label als Ergebnis liefert.

n kann eine positive ganze Zahl oder 0 sein. Wenn *n* gleich 0 ist, ist das Resultat eine leere Zeichenfolge. Wenn *n* größer als die Länge von *Zeichenfolge* ist, gibt @RECHTSB die gesamte *Zeichenfolge* zurück.

Hinweise

@RECHTSB wird für Zwei-Byte-Zeichensätze (DBCS), z. B. Japanisch, verwendet.

@RECHTSB zählt Satz- und Leerzeichen mit.

Wenn es sich beim ersten Byte des zurückgegebenen Labels um die letzte Hälfte eines Zwei-Byte-Zeichens handelt, ersetzt @RECHTSB das unvollständige Zeichen durch ein Leerzeichen.

Verwenden Sie @RECHTSB zusammen mit @FINDENB wenn Sie den genauen Wert von *n* nicht kennen oder wenn *n* möglicherweise variiert.

Beispiele

@RECHTSB("Ein Byte";4) = "Byte"

Ähnliche @Funktionen

@RECHTS liefert die letzten *n* Zeichen aus *Zeichenfolge* zurück. @LINKSB ermittelt die ersten *n* Byte in *Zeichenfolge*. @MITTEB liefert Byte aus einem *Zeichenfolge* zurück.

@SPI

@SPI(*Kapital;Zinssatz;Laufzeit;Periode*) berechnet den Zinsanteil von periodischen Zahlungen bei gleichbleibendem Kapitalanteil pro Zahlungsperiode.

Argumente

Kapital ist der Darlehensbetrag und kann ein beliebiger Wert sein.

Zinssatz ist der Zinssatz für den Kredit. Es kann sich dabei um einen Dezimal- oder einen Prozentwert handeln, der größer als -1 ist.

Laufzeit steht für die Zahl der Zahlungszeiträume. Es kann sich dabei um eine beliebige positive ganze Zahl handeln. Dezimalbrüche werden abgeschnitten.

Zeitraum ist der Zahlungszeitraum, für den Sie die Zinszahlungen bestimmen wollen. *Periode* kann jede beliebige ganze Zahl sein, die kleiner oder gleich *Laufzeit* ist. Dezimalbrüche werden abgeschnitten. *Periode* beginnt bei Null (0).

Hinweise

Der Zeitraum, der für die Berechnung von *Zinssatz* verwendet wird, muß der gleiche sein, der auch für *Laufzeit* verwendet wird. Wenn Sie z.B. die monatlichen Zahlungen berechnen wollen, geben Sie den Zinssatz und die Laufzeit in Monaten ein. Normalerweise heißt dies, daß Sie den Zinssatz durch 12 teilen und die Anzahl der Jahre der *Laufzeit* mit 12 multiplizieren müssen.

Beispiele

Sie haben ein Darlehen von 8000 Dollar bei einem jährlichen Zinssatz von 10,5% und einer Laufzeit von 3 Jahren aufgenommen. Die Tilgung erfolgt monatlich. Den Zinsanteil der ersten Zahlung berechnen Sie folgendermaßen:

$$\text{@SPI}(8000;0,105/12;36;0) = \$70$$

So bestimmen Sie die erste Zahlung:

$$\text{@SPI}(8000;0,105/12;36;0) + 8000/36 = \$292,22$$

Ähnliche @Funktionen

@RATEZ berechnet den Zinsanteil von periodischen Zahlungen, wenn die Zahlungen pro Periode gleich bleiben, der Kapitalanteil aber bei sinkendem Zinssatz steigt.

@THAIDATUMKURZ

@THAIDATUMKURZ(*Datumwert*) konvertiert *Datumwert* in einen Thai-Datumsnamen in kurzem Format.

Argumente

Datumwert ist eine Datumseriennummer.

Beispiele

@THAIDATUMKURZ(@DATUM(91;8;16)) = 16 สค. 34

Ähnliche @Funktionen

@THAIDATUMLANG konvertiert *Datumwert* in einen Thai-Datumsnamen im langen Format.

@THAIZIFFERNFOLGE

@THAIZIFFERNFOLGE(*Ziffernfolge*) konvertiert eine *Ziffernfolge* von arabischen Zahlen in eine Zeichenfolge mit Thai-numerischen Zeichen.

Argumente

Ziffernfolge ist eine Zeichenfolge, die arabische Zahlen enthält.

Beispiele

@THAIZIFFERNFOLGE("16 ๓๓ . 34") = ๑๖ ๓๓ . ๓๔

Ähnliche @Funktionen

@EZIFFERNFOLGE konvertiert eine *Ziffernfolge* von Thai-numerischen Zeichen in arabische Zahlen.

@THAIWOCHENTAG

@THAIWOCHENTAG(*Datumwert*) konvertiert *Datumwert* in einen Thai-Datumsnamen.

Argumente

Datumwert ist eine Datumseriennummer.

Beispiele

@THAIWOCHENTAG(@DATUM(91,8;16)) = ศุกร์

@ANN2

@ANN2(Zahlungen;Zinssatz;Zukwert) berechnet die erforderliche Anzahl von Zahlungsperioden bei gleichbleibenden *Zahlungen* zur Akkumulation von *Zukwert* bei periodischem *Zinssatz* und vorschüssiger Rente.

Argumente

Zahlungen ist der Wert der gleichbleibenden Zahlungen und kann jeder beliebige Wert außer 0 sein.

Zinssatz ist der periodische Zinssatz. Es kann sich dabei um einen Dezimal- oder einen Prozentwert handeln, der größer als -1 ist.

Zukwert ist das Kapital, das Sie sparen wollen. *Zukwert* kann ein beliebiger Wert sein.

Hinweise

Sie können berechnen, welche Laufzeit zur Rückzahlung eines Darlehens notwendig ist, indem Sie **@ANN2** mit einem negativen *Zukwert* verwenden. Sie wollen beispielsweise wissen, wie lange es dauert, ein Darlehen von 10.000 Dollar mit einem Zinssatz von 10% zu tilgen, wenn Sie jährlich 1174 Dollar zurückzahlen:

@ABS(@ANN2(1174,6;0,1;-10000)). Dieser Berechnung zufolge benötigen Sie 16 Jahre für die Rückzahlung des Darlehens.

Anders als **@ANN** verwendet **@ANN2** für die Berechnung nicht den Beginn, sondern das Ende der Laufzeit.

Beispiele

Sie legen am Ende eines jeden Jahres 200.000 Yen auf einem Sparbuch fest. Das Sparbuch hat einen jährlichen Zinssatz von 7,5%. Sie wollen nun berechnen, wie lange Sie brauchen, um 10.000.000 Yen anzusparen:

@ANN2(200000;0,075;10000000) = 20,76 Jahre

Das gleiche Ergebnis würde auch die Formel **@NPER(200000;0,075;10000000;1)** liefern.

Ähnliche @Funktionen

@ANN berechnet die Zahl der erforderlichen Perioden für eine Reihe gleicher Zahlungen zur Akkumulation von *Zukwert* bei gegebenem Zinssatz und bei nachschüssiger Rente. **@NPER** berechnet die Anzahl der erforderlichen Perioden bei gleichbleibenden Zahlungen zur Akkumulation von *Zukwert* bei einem periodischen Zinssatz und bei nachschüssiger oder vorschüssiger Rente.

@LAUF berechnet die Anzahl von Perioden, nach der eine einmalige Investition auf einem bestimmten *Zukwert* anwächst.

@THAIFINDEN

@THAIFINDEN(*Suchfolge*; *Zeichenfolge*; *Startnummer*) berechnet die Spalte des ersten Vorkommens von *Suchfolge* in *Zeichenfolge*, beginnend mit dem Thai-Zeichen bei *Startnummer*. Diese Funktion arbeitet mit logischen Thai-Zeichen, die aus 1 bis 3 Byte bestehen können.

Argumente

Suchfolge und *Zeichenfolge* stehen für durch Anführungsstriche (" ") markierten Text, Formeln, die eine Zeichenfolge zurückgeben, Adressen oder Namen von Zellen, die Label enthalten, oder Formeln, die Label zurückgeben.

Startnummer ist eine Spaltennummer.

Hinweise

@THAIFINDEN ist auch in der Kombination mit @THAIMITTE oder @THAIERSETZEN nützlich, um Text zu finden und zu extrahieren oder zu ersetzen.

Beispiele

@THAIFINDEN("็", "บริษัท ็พชอฝท จำกัด", 0) = 5

Ähnliche @Funktionen

@SUCHEN führt dieselbe Funktion wie **@THAIFINDEN** aus, wird aber für nicht-thailändische Zeichenfolgen verwendet.

@THAIDATUMLANG

@THAIDATUMLANG(*Datumwert*) konvertiert *Datumwert* in einen Thai-Datumsnamen im langen Format.

Argumente

Datumwert ist eine Datumseriennummer.

Beispiele

@THAIDATUMLANG(@DATUM(91;8;16)) = 16 สิงหาคม 2534

Ähnliche @Funktionen

@THAIDATUMKURZ konvertiert *Datumwert* in einen Thai-Datumsnamen im kurzen Format.

@THAILINKS

@THAILINKS(*Zeichenfolge*; *n*) gibt die ersten *n* Spalten in *Zeichenfolge* zurück. Diese Funktion arbeitet mit logischen Thai-Zeichen, die aus 1 bis 3 Byte bestehen können.

Argumente

Zeichenfolge ist durch Anführungsstriche (" ") markierter Text, eine Formel, die eine Zeichenfolge als Ergebnis liefert, die Adresse oder der Name einer Zelle, die ein Label enthält, oder eine Formel, die ein Label als Ergebnis liefert.

n kann eine positive ganze Zahl oder 0 sein. Wenn *n* gleich 0 ist, ist das Resultat eine leere Zeichenfolge. Wenn *n* größer als die Länge von *Zeichenfolge* ist, gibt @THAILINKS die gesamte *Zeichenfolge* zurück.

Beispiele

@THAILINKS("บริษัท วิทยุคอมพ์ จำกัด",4) = บริษัท

Ähnliche @Funktionen

@LINKS führt dieselbe Funktion wie @THAILINKS aus, wird aber für nicht-thailändische Zeichenfolgen verwendet.

@THAIMITTE liefert Thai-Zeichen aus einem *Text* zurück. @THAIRECHTS gibt die letzten *n* Zeichen in *Zeichenfolge* zurück.

@THAILÄNGE

@THAILÄNGE(*Zeichenfolge*) zählt die Spalten in *Zeichenfolge*. Diese Funktion arbeitet mit logischen Thai-Zeichen, die aus 1 bis 3 Byte bestehen können.

Argumente

Zeichenfolge ist durch Anführungsstriche (" ") markierter Text, eine Formel, die eine Zeichenfolge als Ergebnis liefert, die Adresse oder der Name einer Zelle, die ein Label enthält, oder eine Formel, die ein Label als Ergebnis liefert.

Beispiele

@THAILÄNGE("บริษัท วิทยุการบิน จำกัด") = 17

Ähnliche @Funktionen

@LÄNGE führt dieselbe Funktion wie @THAILÄNGE aus, wird aber für nicht-thailändische Zeichenfolgen verwendet.

@THAIMITTE

@THAIMITTE(*Zeichenfolge*; *Startnummer*; *n*) kopiert *n* logische Thai-Zeichen aus *Zeichenfolge*, beginnend mit der Spalte *Startnummer*. Diese Funktion arbeitet mit logischen Thai-Zeichen, die aus 1 bis 3 Byte bestehen können.

Argumente

Zeichenfolge ist durch Anführungsstriche (" ") markierter Text, eine Formel, die eine Zeichenfolge als Ergebnis liefert, die Adresse oder der Name einer Zelle, die ein Label enthält, oder eine Formel, die ein Label als Ergebnis liefert.

Startnummer ist eine Spaltennummer. Wenn *Startnummer* größer als die Länge von *Zeichenfolge* ist, ist das Ergebnis von @THAIMITTE eine leere Zeichenfolge.

n ist eine beliebige positive ganze Zahl oder 0. Wenn *n* gleich 0 ist, ist das Ergebnis von @THAIMITTE eine leere Zeichenfolge. Wenn *n* länger als die Länge von *Zeichenfolge* ist, gibt @THAIMITTE alle Zeichen von der *Startnummer* bis zum Ende von *Zeichenfolge* zurück.

Beispiele

@THAIMITTE("บริษัท ๑๒๓ จำกัด",5,7) = ๑๒๓

Ähnliche @Funktionen

@MITTE führt dieselbe Funktion wie @THAIMITTE aus, wird aber für nicht-thailändische Zeichenfolgen verwendet.

@THAIRECHTS gibt die letzten *n* Zeichen in *Zeichenfolge* zurück. @THAILINKS gibt die ersten *n* Thai-Zeichen in *Zeichenfolge* zurück.

@THAIZAHLENFOLGE

@THAIZAHLENFOLGE(*Zahl*) konvertiert *Zahl* in eine Thai-numerische Zeichenfolge.

Argumente

Zahl ist eine ganze Zahl oder eine Gleitkommazahl.

Beispiele

@THAIZAHLENFOLGE(1000000) = หนึ่งล้านบาทถ้วน

Ähnliche @Funktionen

@ZAHLENFOLGE konvertiert *Zahl* in eine numerische Zeichenfolge (als Label gespeichert) in einem nicht-thailändischen Format.

@THAIERSETZEN

@THAIERSETZEN(*Original-Zeichenfolge*; *Startnummer*; *n*; *Ersatz-Zeichenfolge*) ersetzt *n* Spalten in *Original-Zeichenfolge* durch *Ersatz-Zeichenfolge*, beginnend bei *Startnummer*. Diese Funktion arbeitet mit logischen Thai-Zeichen, die aus 1 bis 3 Byte bestehen können.

Argumente

Original-Zeichenfolge und *Ersatz-Zeichenfolge* sind durch Anführungsstriche (" ") markierter Text, Formeln, die eine Zeichenfolge zurückgeben, die Adressen oder Namen von Zellen, die Label enthalten, oder Formeln, die Label zurückgeben.

Startnummer ist die Spaltennummer in *Original-Zeichenfolge*. Es kann sich dabei um einen beliebigen positiven Wert oder 0 handeln. Wenn *Startnummer* größer als die Länge von *Original-Text* ist, hängt @THAIERSETZEN die *Ersatz-Zeichenfolge* an *Original-Zeichenfolge* an.

n kann eine beliebige positive ganze Zahl oder 0 sein. Wenn *n* gleich 0, fügt @THAIERSETZEN die *Ersatz-Zeichenfolge* bei *Startnummer* ein, ohne Spalten in *Original-Zeichenfolge* zu löschen.

Beispiele

@THAIERSETZEN("บริษัท จำกัด",5,0,"ไทยซอฟต์แวร์") = บริษัท ไทยซอฟต์แวร์ จำกัด

Ähnliche @Funktionen

@ERSETZEN führt dieselbe Funktion wie @THAIERSETZEN aus, wird aber für nicht-thailändische Zeichenfolgen verwendet.

@THAIRECHTS

@THAIRECHTS(*Zeichenfolge*; *n*) gibt die letzten *n* logischen Thai-Zeichen in *Zeichenfolge* zurück. Diese Funktion arbeitet mit logischen Thai-Zeichen, die aus 1 bis 3 Byte bestehen können.

Argumente

Zeichenfolge ist durch Anführungsstriche (" ") markierter Text, eine Formel, die eine Zeichenfolge als Ergebnis liefert, die Adresse oder der Name einer Zelle, die ein Label enthält, oder eine Formel, die ein Label als Ergebnis liefert.

n kann eine positive ganze Zahl oder 0 sein. Wenn *n* gleich 0 ist, ist das Resultat eine leere Zeichenfolge. Wenn *n* größer als die Länge von *Zeichenfolge* ist, gibt @THAIRECHTS die gesamte *Zeichenfolge* zurück.

Beispiele

@THAIRECHTS("บริษัท ไทยซอฟต์แวร์ จำกัด",4) = จำกัด

Ähnliche @Funktionen

@RECHTS führt dieselbe Funktion wie @THAIRECHTS aus, wird aber für nicht-thailändische Zeichenfolgen verwendet. @THAIMITTE liefert Thai-Zeichen aus *Zeichenfolge* zurück. @THAILINKS gibt die ersten *n* Thai-Zeichen in *Zeichenfolge* zurück.

@XINTZINS

@XINTZINS(*Schätzwert;Cash-flow;Datum*) berechnet die interne Ertragsrate für eine Reihe von Einnahmen und Ausgaben.

Argumente

Schätzwert ist ein Dezimal- oder Prozentwert, der die geschätzte interne Ertragsrate angibt. In den meisten Fällen sollte der Schätzwert zwischen 0 (0%) und 1 (100%) liegen. 10% ist häufig ein guter Schätzwert. Bei sehr hohen Cash-flows sollte der *Schätzwert* so genau wie möglich sein.

Cash-flow ist die Adresse bzw. der Name eines Bereichs, der den Cash-flow enthält. 1-2-3 betrachtet negative Zahlen als Auszahlungsströme und positive Zahlen als Einzahlungsströme. Normalerweise ist der erste Cash-flow-Betrag im Bereich eine negative Zahl (ein Auszahlungsstrom), die eine Investition darstellt. 1-2-3 weist allen leeren Zellen und Labeln im Bereich den Wert 0 zu und nimmt sie in die Berechnung auf.

Datum ist die Adresse oder der Name eines Bereichs, der das Datum angibt, an dem die entsprechenden Cash-flows stattfinden. Jedes Datum kann eine beliebige Datumseriennummer sein und entspricht dem Zeitpunkt der entsprechenden Bewegung im *Cash-flow*-Bereich. Die Datumseriennummern müssen in aufsteigender Reihenfolge vorliegen.

Die Bereiche für *Cash-flow* und *Datum* müssen gleich groß sein, d.h., sie müssen die gleiche Zellenanzahl enthalten. Zellen in zwei Bereichen werden anhand ihrer Reihenfolge im Bereich zu Paaren zusammengefaßt. Bereiche werden von oben nach unten (Zeilen nach unten), dann von links nach rechts (über Spalten) und anschließend von Arbeitsblatt zu Arbeitsblatt geordnet. Wenn die Bereiche für *Cash-flow* und *Datum* nicht gleich groß sind, liefert @XINTZINS einen FEHLER.

Hinweise

Cash-flows können in unregelmäßigen Intervallen vorgenommen werden. Jeder Cash-flow in *Cash-flow* wird mit einem Datum in *Datum* zu einem Paar zusammengefaßt. Der erste Cash-flow und das erste Datum kennzeichnen den Anfang des Plans; der erste Cash-flow wird nicht diskontiert. Spätere Cash-flows werden ausgehend vom jährlichen Diskontsatz und dem durch das entsprechende Datum angezeigten Zeitpunkt der Bewegung diskontiert.

@INTZINS diskontiert den ersten Cash-flow. @XINTZINS tut dies nicht und verwendet das erste Datum für den Beginn des Plans. Verwenden Sie 0 für den ersten Cash-flow in @XINTZINS, um die Diskontierkonvention von @INTZINS zu kopieren.

@XINTZINS erlaubt Cash-flows in unregelmäßigen Intervallen. @INTZINS geht von regelmäßigen Intervallen aus.

Beispiele

Sie investieren am 13. September 1996 50.000 Dollar und erhalten die folgenden Zahlungen: 1000 Dollar am 31. Januar 1997 und 53.000 Dollar am 14. Juni 1997. Der folgende Bereich zeigt diese Daten:

| SCHÄTZWERT | CASH-FLOW | DATUM |
|------------|-----------|----------|
| 0,10 | -50000 | 13/09/96 |
| | 1000 | 31/01/97 |
| | 53000 | 14/06/97 |

@XINTZINS(SCHÄTZWERT;CASH-FLOW;DATUM) = 10,90%

Ähnliche @Funktionen

@XNETAKTWERT berechnet den aktuellen Nettowert einer Reihe von Einnahmen und Ausgaben. @INTZINS berechnet die interne Ertragsrate (Profit) für eine Reihe von Cash-flow-Werten, die durch eine Investition ausgelöst wurden. @MINTZINS berechnet die geänderte interne Ertragsrate.

@XNETAKTWERT

@XNETAKTWERT(*Zinssatz*; *Cash-flow*; *Datum*) berechnet den aktuellen Nettowert einer Reihe von Einnahmen und Ausgaben.

Argumente

Zinssatz ist der Diskontsatz und kann ein beliebiger Wert sein, der größer als -1 ist.

Cash-flow ist die Adresse bzw. der Name eines Bereichs, der den Cash-flow enthält. 1-2-3 betrachtet negative Zahlen als Auszahlungsströme und positive Zahlen als Einzahlungsströme. Normalerweise ist der erste Cash-flow-Betrag im Bereich eine negative Zahl (ein Auszahlungsstrom), die eine Investition darstellt. 1-2-3 weist allen leeren Zellen und Labeln im Bereich den Wert 0 zu und nimmt sie in die Berechnung auf.

Datum ist die Adresse oder der Name eines Bereichs, der das Datum angibt, an dem die entsprechenden Cash-flows stattfinden. Jedes Datum kann eine beliebige Datumseriennummer sein und entspricht dem Zeitpunkt der entsprechenden Bewegung im *Cash-flow*-Bereich. Die Datumseriennummern müssen in aufsteigender Reihenfolge vorliegen.

Die Bereiche für *Cash-flow* und *Datum* müssen gleich groß sein, d.h., sie müssen die gleiche Zellenanzahl enthalten. Zellen in zwei Bereichen werden anhand ihrer Reihenfolge im Bereich zu Paaren zusammengefaßt. Bereiche werden von oben nach unten (von Zeile zu Zeile), dann von links nach rechts (von Spalte zu Spalte) und anschließend von Arbeitsblatt zu Arbeitsblatt geordnet. Wenn die Bereiche für *Cash-flow* und *Datum* nicht gleich groß sind, liefert @XNPV einen FEHLER.

Hinweise

Cash-flows können in ungleichen Intervallen vorgenommen werden. Jeder Cash-flow in *Cash-flow* wird mit einem Datum in *Datum* zu einem Paar zusammengefaßt. Der erste Cash-flow und das erste Datum kennzeichnen den Anfang der Tabelle; der erste Cash-flow wird nicht diskontiert. Spätere Cash-flows werden ausgehend vom jährlichen Diskontsatz und dem durch das entsprechende Datum angezeigten Zeitpunkt der Bewegung diskontiert.

Beispiele

In diesem Beispiel wird @XNETAKTWERT verwendet, um den aktuellen Wert einer Reihe von ungleichen, unregelmäßigen Zahlungen zu diskontieren, die bei einem jährlichen Zinssatz von 11,5% investiert werden. In dem Beispiel werden die folgenden Daten verwendet: *Cash-flow* ist eine Liste von Cash-flows im Bereich ZAHLUNGEN, und *Datum* ist die Liste mit den Angaben zum Zeitpunkt der Cash-flows, gespeichert in dem Bereich DATUM:

| ZAHLUNGEN | DATUM |
|--------------|----------|
| \$0,00 | 01.01.96 |
| \$250.000,00 | 01.04.96 |
| \$250.000,00 | 01.05.96 |
| \$300.000,00 | 01.06.96 |
| \$500.000,00 | 01.07.96 |
| \$600.000,00 | 01.08.96 |
| \$900.000,00 | 01.09.96 |
| \$300.000,00 | 01.10.96 |
| \$250.000,00 | 01.11.96 |
| \$750.000,00 | 01.01.97 |

@XNETAKTWERT(0,115;ZAHLUNGEN;DATUM) = \$3.821.809,20

Ähnliche @Funktionen

@NETAKTWERT berechnet den aktuellen Nettowert einer Kapitalanlage, indem es den aktuellen Wert vom zukünftigen Wert diskontiert. @XINTZINS berechnet die interne Ertragsrate einer Reihe von Einnahmen und Ausgaben.

@RENDITE2

@RENDITE2(*Valutierungstermin*; *Fälligkeit*; *Coupon*; *Kurs*; [*Rückzahlung*]; [*Zinstage*]) berechnet die Rendite für Wertpapiere mit periodischen Zinszahlungen auf der Grundlage japanischer Konventionen.

Argumente

Valutierungstermin ist der Abrechnungstermin für das Wertpapier. *Valutierungstermin* ist eine Datumseriennummer.

Fälligkeit ist das Fälligkeitsdatum des Wertpapiers. *Fälligkeit* ist eine Datumseriennummer. Wenn *Fälligkeit* kleiner als oder gleich *Valutierungstermin*, liefert @RENDITE2 einen FEHLER.

Coupon ist der jährliche Zinssatz des Wertpapiers, angegeben als Dezimalwert. *Coupon* ist ein beliebiger positiver Wert oder 0.

Kurs ist der Kurs in Yen für das Wertpapier pro 100 Yen Nennwert. *Kurs* kann ein beliebiger positiver Wert sein.

Rückzahlung ist ein wahlfreies Argument, das den Rückzahlungswert des Wertpapiers pro ¥100 Nennwert angibt.

redemption ist ein beliebiger positiver Wert oder 0. Wenn Sie das Argument *Rückzahlung* weglassen, verwendet @RENDITE2 den Wert 100.

Zinstage ist ein wahlfreies Argument, das die Art der verwendeten Berechnungsbasis angibt:

| <u>Zinstage</u> | <u>Berechnungsbasis</u> |
|-----------------|-------------------------|
| 0 | 30/360 |
| 1 | Anzahl/Anzahl |
| 2 | Anzahl/360 |
| 3 | Anzahl/365 |
| 4 | Europäisch 30/360 |

Wenn Sie das Argument *Zinstage* verwenden, ruft 1-2-3 die Funktion @RENDITE auf, um den Wert zu berechnen.

Beispiele

Eine Anleihe hat den Valutierungstermin 1. Juli 1993. Das Fälligkeitsdatum ist der 1. Dezember 1998. Die halbjährliche Zinsrate beträgt 5,50%. Der Kurs der Anleihe ist 99,5 Yen. Der Rückzahlungswert beträgt 100 Yen. So berechnen Sie die Rendite der Anleihe:

@RENDITE2(@DATUM(93;7;1),@DATUM(98;12;1),0,055;99,5;100) = 0,056072

Ähnliche @Funktionen

@RENDITE berechnet die Rendite für Wertpapiere mit periodischen Zinszahlungen.

@STÜCKZINS berechnet den Stückzins für Wertpapiere mit periodischen Zinszahlungen unter Verwendung japanischer Konventionen. @KURS2 berechnet den Kurs pro 100 Yen Nennwert für Wertpapiere mit periodischen Zinszahlungen auf der Grundlage japanischer Konventionen.

@STÜCKZINS berechnet die Stückzinsen für Wertpapiere mit periodischen Zinszahlungen. @KURS berechnet den Kurs pro 100 Dollar Nennwert für Wertpapiere mit periodischen Zinszahlungen auf der Grundlage japanischer Konventionen. @LAUFZEIT berechnet die jährliche Laufzeit für Wertpapiere mit periodischen Zinszahlungen.

@MLAUFZEIT berechnet die modifizierte jährliche Laufzeit für Wertpapiere mit periodischen Zinszahlungen.

