

## 12. Übersicht der Funktionen

Hier finden Sie alle in List & Label verfügbare Funktionen als alphabetische Liste. Im Formel-Assistenten steht Ihnen zusätzlich eine nach Funktionsgruppe sortierte Liste zur Verfügung.

### Abs

---

**Aufgabe:**

Gibt den Absolutwert einer Zahl zurück, d.h. eine evtl. negative Zahl wird positiv zurückgegeben, eine positive Zahl bleibt unverändert.

**Parameter:**

Zahl

**Rückgabewert:**

Zahl

**Beispiele:**

Abs (-3)

Ergebnis: 3

Abs (3.12)

Ergebnis: 3.12

### AddDays

---

**Aufgabe:**

Addiert die angegebene Anzahl von Tagen zu dem Datum hinzu, bzw. zieht diese bei negativen Werten ab.

**Parameter:**

Datum

Zahl

**Rückgabewert:**

Datum

**Beispiele:**

AddDays(Date("04.07.1776"),3)

Ergebnis: 07.07.1776

AddDays(Date("04.07.1776"),-3)

Ergebnis: 01.07.1776

### AddHours

---

**Aufgabe:**

Addiert die angegebene Anzahl von Stunden zu dem Datum hinzu, bzw. zieht diese bei negativen Werten ab.

**Parameter:**

Datum

Zahl

Rückgabewert:

Datum

Beispiele:

AddHours(Date("04.07.1776"),48)                      Ergebnis: 06.07.1776

AddHours(Date("04.07.1776"),-48)                      Ergebnis: 02.07.1776

---

## AddMinutes

Aufgabe:

Addiert die angegebene Anzahl von Minuten zu dem Datum hinzu, bzw. zieht diese bei negativen Werten ab.

Parameter:

Datum

Zahl

Rückgabewert:

Datum

Beispiele:

AddMinutes(Date("04.07.1776"),2880)                      Ergebnis: 06.07.1776

AddMinutes(Date("04.07.1776"),-2880)                      Ergebnis: 02.07.1776

---

## AddMonths

Aufgabe:

Addiert die angegebene Anzahl von Monaten zu dem Datum hinzu, bzw. zieht diese bei negativen Werten ab.

Parameter:

Datum

Zahl

Rückgabewert:

Datum

Beispiele:

AddMonths(Date('04.07.1776'),3)                      Ergebnis: 04.10.1776

AddMonths(Date('04.07.1776'),-3)                      Ergebnis: 04.04.1776

---

## AddSeconds

Aufgabe:

Addiert die angegebene Anzahl Sekunden zu dem Datum hinzu, bzw. zieht diese bei negativen Werten ab.

**Parameter:**

Datum

Zahl

**Rückgabewert:**

Datum

**Beispiele:**

AddSeconds(Date("04.07.1776"),172800)                      Ergebnis: 06.07.1776

AddSeconds(Date("04.07.1776"),-172800)                      Ergebnis: 02.07.1776

### AddWeeks

---

**Aufgabe:**

Addiert die angegebene Anzahl von Wochen zu dem Datum hinzu, bzw. zieht diese bei negativen Werten ab.

**Parameter:**

Datum

Zahl

**Rückgabewert:**

Datum

**Beispiele:**

AddWeeks(Date("04.07.1776"),4)                              Ergebnis: 01.08.1776

AddWeeks(Date("04.07.1776"),-4)                              Ergebnis: 06.06.1776

### AddYears

---

**Aufgabe:**

Addiert die angegebene Anzahl von Jahren zu dem Datum hinzu, bzw. zieht diese bei negativen Werten ab.

**Parameter:**

Datum

Zahl

**Rückgabewert:**

Datum

**Beispiele:**

AddYears(Date("04.07.1776"),4)                              Ergebnis: 04.07.1776

AddYears(Date("04.07.1776"),-4)                              Ergebnis: 04.07.1772

## Alias\$

---

### Aufgabe:

Gibt den Wert zurück, der für den Schlüssel (erster Parameter), bei den Schlüssel/Wertepaaren (zweiter Parameter) angegeben wird.

### Parameter:

**Zeichenkette** Ausdruck für den zu suchenden Wert.

**Zeichenkette** Liste der Werte (Form: <Schlüsselwert> | [<Schlüsselwert>]. Um "|" oder "=" im Wert oder Schlüssel verwenden zu können, stellen Sie diesen ein "\" voran.

**Zeichenkette** (optional) Voreinstellung wenn der Wert nicht gefunden werden kann.

### Rückgabewert:

**Zeichenkette**

### Beispiel:

Alias\$ ("DEU", "DEU=Deutschland|USA=United States of America|GB=United Kingdom")  
Ergebnis: "Deutschland"

## ArcCos

---

### Aufgabe:

Berechnet den Arkus-Kosinus des Werts (in Grad, wenn keine andere Einheit gewählt ist).

### Parameter:

**Zahl** Wert

**Zahl** (optional) Modus (0=Grad, 1=Radian). Voreinstellung: 0.

### Rückgabewert:

**Zahl**

### Beispiel:

ArcCos (0) Ergebnis: 90

## ArcSin

---

### Aufgabe:

Berechnet den Arkus-Sinus des Werts (in Grad, wenn keine andere Einheit gewählt ist).

### Parameter:

**Zahl** Wert

**Zahl** (optional) Modus (0=Grad, 1=Radian). Voreinstellung: 0.

### Rückgabewert:

### Zahl

#### Beispiel:

ArcSin (0.5)

Ergebnis: 30,00

## ArcTan

---

#### Aufgabe:

Berechnet den Arkus-Tangens des Werts (in Grad, wenn keine andere Einheit gewählt ist).

#### Parameter:

Zahl Wert

Zahl (optional) Modus (0=Grad, 1=Radiant). Voreinstellung: 0.

#### Rückgabewert:

Zahl

#### Beispiel:

ArcTan (1)

Ergebnis: 45,00

## Asc

---

#### Aufgabe:

Gibt den ASCII-Code des ersten Zeichens der Zeichenkette zurück.

#### Parameter:

Zeichenkette

#### Rückgabewert:

Zahl

#### Beispiel:

Asc ("A")

Ergebnis: 65

## AskString\$

---

#### Aufgabe:

Hierüber können während der Laufzeit des Drucks Informationen vom Anwender abgefragt werden. Ein typisches Anwendungsbeispiel wäre das Projekt eines Überweisungsformulars. Feststehende Angaben wie Name und Bankverbindungen des Absenders lassen sich über festen Text oder Variablen direkt in das Projekt integrieren. Die zu überweisenden Beträge werden sich jedoch von Fall zu Fall unterscheiden und können daher schlecht im Projekt vorgegeben werden. Über die Funktion AskString\$() kann jedoch eine solche Information während des Drucks vom Anwender erfragt werden.

Beim Druck erscheint dann eine Dialogabfrage, in der die gewünschte Information einzugeben ist.

Der Dialog erlaubt den eingegebenen Wert entweder zu übernehmen (Button OK) oder die Funktion abzubrechen (Button Abbrechen). Das Abbrechen der Funktion bedeutet, dass der Druckvorgang abgebrochen wird.

Über die Schaltfläche "Alles" können alle noch ausstehenden Benutzereingaben automatisch durch den aktuell eingegebenen Wert beantwortet werden. Dies ist hilfreich, wenn der Dialog für jeden einzelnen Datensatz erscheint, jedoch für alle Datensätze derselbe Wert eingegeben werden soll.

**Parameter:**

- Zeichenkette** Text, der als Eingabeaufforderung in dem Dialog zur Benutzereingabe erscheinen soll. Dieser erste Parameter muss angegeben werden, die übrigen Parameter sind optional. Falls keine weiteren Parameter angegeben werden, ist der erste Parameter zugleich der Vorschlagswert für die Benutzereingabe.
- Boolean** (optional) Gibt an, ob die Benutzerabfrage nur einmal zu Beginn des Drucks erscheinen soll (False, Voreinstellung), oder ob die Information für jeden Datensatz individuell abgefragt werden soll (True).
- Zeichenkette** (optional) Der Vorschlagswert im Eingabefeld des Dialogs.
- Zahl** (optional) Gibt die Anzahl der Stellen an, die als Benutzereingabe zulässig sein soll. Ein Wert von 8 beispielsweise bedeutet, dass der Benutzer maximal 8 Stellen eingeben kann.

**Rückgabewert:**

**Zeichenkette**

**Beispiele:**

`AskString$ ("Überweisungsbetrag", True, "50,00 EUR", 8)`

Öffnet eine Dialogbox mit dem Titel "Überweisungsbetrag", dem vorgeschlagenen Wert "50,00 EUR" und zulässigen Stellenanzahl von 8 Stellen für die Eingabe. Da der zweite Parameter True lautet, erscheint diese Abfrage für jeden Datensatz im Druck.

---

## **AskStringChoice\$**

**Aufgabe:**

Wie AskString\$, jedoch wird der Wert über einen Dialog mit Combobox abgefragt.

**Parameter:**



**Boolean** (optional) True: nach der Ausgabe werden die für die Berechnung gemerkten Werte gelöscht (Voreinstellung: True). Bitte beachten Sie, dass die für die Berechnung gemerkten Werte generell bei jedem (Unter-)Tabellenende gelöscht werden. Der zweite Parameter bestimmt lediglich, ob die Werte bei einer Ausgabe schon innerhalb der Tabelle gelöscht werden.

**Rückgabewert:**

**Zahl**

**Beispiel:**

Avg (Anzahl \* Stückpreis)

Ergebnis: Mittelwert Gesamtpreis

---

## Barcode

**Aufgabe:**

Wandelt eine Zeichenkette in einen Barcode um.

**Parameter:**

**Zeichenkette** Barcodewert (Inhalt)

**Zeichenkette** Barcodetyp. Die möglichen Barcodetypen werden Ihnen von der Auto-Vervollständigung des Assistenten angeboten. Wenn der Barcode nicht korrekt interpretiert werden kann, wird er auch nicht gedruckt. Für einige Barcodes gibt es bestimmte Formatierungen, die eingehalten werden müssen. Nähere Informationen finden Sie im Kapitel "Barcode-Objekte".

**Rückgabewert:**

**Barcode**

**Beispiel:**

Barcode ("Hallo Welt","GS1 128")

---

## Barcode\$

**Aufgabe:**

Gibt den Textinhalt eines Barcodes zurück.

**Parameter:**

**Barcode**

**Rückgabewert:**

**Zeichenkette**

**Beispiel:**

Barcode\$ (Artikel.Nr\_EAN128)

Ergebnis: "4711"

## BarcodeType\$

---

### Aufgabe:

Gibt den Typ des Barcodes als Zeichenkette zurück.

### Parameter:

Barcode

### Rückgabewert:

Zeichenkette

### Beispiel:

BarcodeType\$ (Artikel.Nr\_EAN128)                      Ergebnis: "EAN 128"

## BasedStr\$

---

### Aufgabe:

Gibt den Wert zu einer beliebigen Basis aus.

### Parameter:

Zahl                      Wert.

Zahl                      Basis (2 bis 36).

Zahl                      (optional) Mindestlänge der Zeichenkette (ohne optionalem Präfix). 0 für die minimale Länge (Voreinstellung).

Boolean                      (optional) gibt an, ob ein Präfix ('0b' für Basis 2, '0o' für Basis 8, '0x' für Basis 16) vor der Zeichenkette eingefügt werden soll (Voreinstellung: False).

### Rückgabewert:

Zeichenkette

### Beispiel:

BasedStr\$ (1,2,1,True)                      Ergebnis: "0b1"

## BinaryAND

---

### Aufgabe:

Verknüpft die beiden (ganzzahligen) Parameter binär mit 'und' und gibt das Resultat zurück.

### Parameter:

Zahl                      Wert.

Zahl                      Wert.

### Rückgabewert:

Zahl

### Beispiel:

BinaryAND (01,10)	Ergebnis: 0
BinaryAND (10,11)	Ergebnis: 10

## BinaryNOT

---

### Aufgabe:

Negiert den Wert binär und gibt das Resultat zurück.

### Parameter:

Zahl                      Wert.

### Rückgabewert:

Zahl

### Beispiel:

BinaryNOT (10)	Ergebnis: 5 (Zehn = 1010, Fünf = 0101)
----------------	--

## BinaryOR

---

### Aufgabe:

Verknüpft die beiden (ganzzahligen) Parameter binär mit 'oder' und gibt das Resultat zurück.

### Parameter:

Zahl                      Wert.

Zahl                      Wert.

### Rückgabewert:

Zahl

### Beispiel:

BinaryOR (01,10)	Ergebnis: 11
------------------	--------------

BinaryOR (10,11)	Ergebnis: 11
------------------	--------------

## BinarySHL

---

### Aufgabe:

Verschiebt den Wert binär nach links.

### Parameter:

Zahl                      Wert.

Zahl                      Zahl der Bits, um die der Wert verschoben wird.

### Rückgabewert:

Zahl

**Beispiel:**

BinarySHL (10,1)                      Ergebnis: 20

## **BinarySHR**

---

**Aufgabe:**

Verschiebt den Wert binär nach rechts.

**Parameter:**

**Zahl**                      Wert.

**Zahl**                      Zahl der Bits, um die der Wert verschoben wird.

**Rückgabewert:**

**Zahl**

**Beispiel:**

BinarySHR (10,1)                      Ergebnis: 5

## **BinaryXOR**

---

**Aufgabe:**

Verknüpft die beiden (ganzzahligen) Parameter binär mit 'exklusiv oder' und gibt das Resultat zurück.

**Parameter:**

**Zahl**                      Wert.

**Zahl**                      Wert.

**Rückgabewert:**

**Zahl**

**Beispiel:**

BinaryXOR (01,10)                      Ergebnis: 11

BinaryXOR (10,11)                      Ergebnis: 1

## **BMPMapToGray**

---

**Aufgabe:**

Wandelt das Bild in Graustufen.

**Parameter:**

**Bild oder Zeichenkette**

**Rückgabewert:**

**Bild**

**Beispiel:**

BMPMapToGray ("sunshine.gif")

## **BMPRotate**

---

### **Aufgabe:**

Dreht ein Bild um die angegebene Gradzahl.

### **Parameter:**

**Bild oder Zeichenkette**

**Zahl** Drehwinkel

**Zahl** (optional) Modus (0=Grad, 1=Radian)

### **Rückgabewert:**

**Bild**

### **Beispiel:**

BMPRotate (Artikel.Bild,90)

## **Capitalize\$**

---

### **Aufgabe:**

Gibt eine Zeichenkette zurück, in der bei den einzelnen Wörtern der erste Buchstabe groß-, der Rest kleingeschrieben ist.

### **Parameter:**

**String**

### **Rückgabewert:**

**Zeichenkette**

### **Beispiel:**

Capitalize\$ (Produkt.Kategorie) Ergebnis: "Tee, Kaffee, Und Softdrinks"

## **Case\$**

---

### **Aufgabe:**

Wandelt eine Zahl je nach Wert in eine Zeichenkette um. Die Zuordnung wird über eine Formatierungs-Zeichenkette gemacht, der die zu ersetzende Zeichenkette für Zahlenwerte von 0 aufsteigend enthält.

### **Parameter:**

**Zahl** Umzuwandelnde Zahl (n). In die Rückgabezeichenkette wird nun der n-te Wert des Zeichenkettfeldes kopiert, wenn nicht genug Werte existieren, ist die Zeichenkette leer.

**Zeichenkette** Ansammlung von verketteten Zeichenketten, die durch ein bestimmtes Zeichen getrennt sind. Wenn kein 3. Parameter

existiert, ist dies "|", ansonsten das erste Zeichen dieses Parameters.

**Zeichenkette** (optional) Trennzeichen für die Formatierungs-Zeichenkette (Voreinstellung: "|").

**Rückgabewert:**

**Zeichenkette**

**Beispiel:**

Case\$(4,"0|I|II|III|IV|V|VI|VII|VIII|IX|X")      Ergebnis: "III"

## Ceil

---

**Aufgabe:**

Berechnet ausgehend vom übergebenen Wert die nächstgrößere Ganzzahl. Siehe auch Funktion Floor().

**Parameter:**

**Zahl**

**Rückgabewert:**

**Zahl**

**Beispiel:**

Ceil (5.6)      Ergebnis: 6

## Century

---

**Aufgabe:**

Bestimmt das Jahrhundert eines Datums und gibt es als Zahl zurück.

**Parameter:**

**Datum**

**Boolean** (optional) Bestimmt, ob die Berechnung "einfach" (Jahrhundert beginnt mit Jahr 0) oder historisch (Jahrhundert beginnt mit Jahr 1) durchgeführt werden soll. Voreinstellung: False (historisch).

**Rückgabewert:**

**Zahl**

**Beispiel:**

Century (Date("04.07.1776"))      Ergebnis: 18

## CheckMod10

---

**Aufgabe:**

Gibt die Modulo 10 Prüfziffer der übergebenen Zeichenkette zurück.

**Parameter:**

**Zeichenkette**

**Rückgabewert:**

**Zahl**

**Beispiel:**

CheckMod10 ("1010101")

Ergebnis: 5

---

## Chr\$

**Aufgabe:**

Wandelt eine Zahl in eine Zeichenkette mit einem Zeichen. Dieses Zeichen hat den ANSI-Code der angegebenen Zahl. Bei Multibyte-Zeichensätzen gilt: HIBYTE=lead byte, LOBYTE= trail byte. Bei Unicode ist der Wert der Zeichenwert (code point).

**Parameter:**

**Zahl**

**Zahl**

(optional) Definiert den Typ des Parameters. 0=Multibyte-Zeichensatz, 1=Unicode. Voreinstellung ist Unicode.

**Rückgabewert:**

**Zeichenkette**

**Beispiel:**

Chr\$ (64)

Ergebnis: "@"

---

## ChrSubst\$

**Aufgabe:**

Untersucht eine Zeichenkette auf Zeichen, die im zweiten Parameter angegeben sind. Jedes Vorkommen eines dieser Zeichen wird durch die komplette Zeichenkette im dritten Parameter ersetzt. Wenn kein dritter Parameter existiert, werden die Zeichen entfernt.

**Parameter:**

**Zeichenkette**

**Zeichenkette**

**Zeichenkette** (optional)

**Rückgabewert:**

**Zeichenkette**

**Beispiel**

ChrSubst\$ ("Hallo Welt", "l", "\_")

Ergebnis: "Ha\_\_o We\_t"

ChrSubst\$ ("Hallo Welt", "aeiou", "\_")

Ergebnis: "H\_ll\_W\_lt"

## Cond

---

### Aufgabe:

Erlaubt das Formulieren beliebiger Bedingungen. Das erste Argument ist ein logischer Ausdruck, der auf seinen Wahrheitsgehalt geprüft wird. Ist der Ausdruck wahr (True), so wird das zweite Argument als Ergebnis zurückgegeben. Ist der Ausdruck falsch (False), so wird das dritte Argument als Ergebnis zurückgegeben. Ist kein drittes Argument angegeben, nimmt der Rückgabewert bei False in Abhängigkeit vom Wertetyp des zweiten Argumentes folgende Standardwerte an:

2. Argument vom Typ	Rückgabewert bei Bedingung = False
Boolean	False
Zeichenkette	"" (leere Zeichenkette)
Datum	Julianischer Datumswert 0
Zahl	0
Bild	"" (leere Zeichenkette)
Barcode	"" (leere Zeichenkette)

### Parameter:

**Boolean**

**Alle**

**Alle**

(optional) Das dritte Argument muss vom gleichen Typ wie das zweite Argument sein.

### Rückgabewert:

**Alle**

### Beispiel:

Cond (Even(5), "gerade", "ungerade")

Ergebnis: ungerade

## Constant.Pi

---

### Aufgabe:

Gibt den Wert der Kreiszahl Pi zurück.

### Parameter: -

### Rückgabewert:

**Zahl**

### Beispiel:

Constant.Pi ()

Ergebnis: 3,14159 (je nach Anzahl der Dezimalstellen)

## Contains

---

**Aufgabe:**

Überprüft, ob eine Zeichenkette eine andere Zeichenkette (zweiter Parameter) enthält.

**Parameter:**

Zeichenkette

Zeichenkette

**Rückgabewert:**

Boolean

**Beispiel:**

Contains ("Hallo", "ll")

Ergebnis: True

## Continued

---

**Aufgabe:**

Gibt an, dass ein Text- oder RTF-Objekt einen Seitenumbruch hatte, d.h. die Ausgabe jetzt eine Folge des Seitenumbruchs ist.

**Parameter: -**

**Rückgabewert:**

Boolean

## Cos

---

**Aufgabe:**

Berechnet den Cosinus des Werts (in Grad, wenn keine andere Einheit gewählt ist).

**Parameter:**

Zahl                      Wert

Zahl                      (optional) Modus (0=Grad, 1=Radian). Voreinstellung: 0.

**Rückgabewert:**

Zahl

**Beispiel:**

Cos (90)

Ergebnis: 0

## Count

---

**Aufgabe:**

Zählt die Anzahl der Werte des Arguments. Aggregatsfunktionen arbeiten grundsätzlich tabellenspezifisch.

Bei dieser Funktion werden Null-Werte des Arguments zur Anzahl hinzugezählt. Verwenden Sie die Funktion `CountIf()`, wenn Sie Null-Werte ausschließen möchten.

**Hinweis:** Summenvariablen (siehe Kapitel "Arbeiten mit Summenvariablen") sind eine alternative Möglichkeit Summen und Zähler zu bilden. Summenvariablen arbeiten grundsätzlich tabellenübergreifend.

### Parameter:

- Alle** Zu zählender Wert (bestimmt die zu zählenden Werte). Erforderlich um die (Unter-)Tabelle zu definieren, für die die Werte gezählt werden.
- Boolean** (optional) True: nach der Ausgabe werden die für die Berechnung gemerkten Werte gelöscht (Voreinstellung: True). Bitte beachten Sie, dass die für die Berechnung gemerkten Werte generell bei jedem (Unter-)Tabellenende gelöscht werden. Der zweite Parameter bestimmt lediglich, ob die Werte bei einer Ausgabe schon innerhalb der Tabelle gelöscht werden.

### Rückgabewert:

Zahl

### Beispiel:

Count (LL.FCountData) Ergebnis: 1 (1 gedruckter Datensatz)

## CountIf

---

### Aufgabe:

Zählt die Anzahl der Werte, für die die Bedingung zutrifft. Verwenden Sie zusätzlich die Funktion `Distinct()`, wenn mehrfach vorkommende Werte nur einmal in die Berechnung eingehen sollen.

### Parameter:

- Boolean** Ausdruck für den Vergleich.
- Boolean** (optional) True: nach der Ausgabe werden die für die Berechnung gemerkten Werte gelöscht (Voreinstellung: True). Bitte beachten Sie, dass die für die Berechnung gemerkten Werte generell bei jedem (Unter-)Tabellenende gelöscht werden. Der zweite Parameter bestimmt lediglich, ob die Werte bei einer Ausgabe schon innerhalb der Tabelle gelöscht werden.

### Rückgabewert:

Zahl

### Beispiel:

- CountIf (Customers.Region="D")  
Zählt alle Werte aus "D"
- CountIf (Distinct(Customers.Region="D"))  
zählt mehrfach vorkommende Werte nur einmal
- CountIf (IsNull (Orders.OrderDate))  
zählt alle Werte mit leerem Feldinhalt.

### **Crosstab.Cells.Avg**

---

**Aufgabe:**

Gibt (in Kreuztabellen) den Mittelwert der Zelleninhalte zurück.

**Parameter:**

- Boolean** (optional) True: Nur definierte Werte gehen in die Berechnung ein (Voreinstellung: False). Definierte Werte: Wenn Sie z.B. eine Auswertung über Kunden und Quartale erstellen, entsprechen die Quartale ohne Umsatz einem "nicht definierten Wert" und können bei der Berechnung extra behandelt werden.
- Zahl** (optional) Zeilen-Ebene (0= unterste Ebene bzw. innerste Gruppierung, 1 = nächsthöhere, ...). Voreinstellung: 0.
- Zahl** (optional) Spalten-Ebene (0= unterste Ebene bzw. innerste Gruppierung, 1 = nächsthöhere, ...). Voreinstellung: 0.

**Rückgabewert:**

Zahl

**Beispiel:**

Crosstab.Cells.Avg (True,2,0)                      Ergebnis: Mittelwert der Zeile

### **Crosstab.Cells.Max**

---

**Aufgabe:**

Gibt (in Kreuztabellen) den größten Wert der Zelleninhalte zurück.  
Parameter und Rückgabewert siehe Funktion Crosstab.Cells.Avg()

**Beispiel:**

Crosstab.Cells.Max (True,2,0)                      Ergebnis: Maximalwert der Zeile

### **Crosstab.Cells.Min**

---

**Aufgabe:**

Gibt (in Kreuztabellen) den kleinsten Wert der Zelleninhalte zurück.  
Parameter und Rückgabewert siehe Funktion Crosstab.Cells.Avg()

**Beispiel:**

Crosstab.Cells.Min (True,2,0)

Ergebnis: Minimalwert der Zeile

---

**Crosstab.Col\$**

**Aufgabe:**

Gibt (in Kreuztabellen) den Spaltenbezeichner für die momentan auszugebende Zelle zurück.

**Parameter:**

**Zahl** (optional) Spalten-Ebene (0= unterste Ebene bzw. innerste Gruppierung, 1= nächsthöhere, ...). Voreinstellung: 0.

**Rückgabewert:**

Zeichenkette

---

**Crosstab.Col**

**Aufgabe:**

Gibt (in Kreuztabellen) den Spaltenindex für die momentan auszugebende Zelle zurück. Die erste Spalte hat den Index 0.

**Parameter:**

**Boolean** (optional) True: Nur Zellen der gleichen Ebene zählen (Voreinstellung False).

**Rückgabewert:**

Zahl

---

**Crosstab.Row\$**

**Aufgabe:**

Gibt (in Kreuztabellen) den Zeilenbezeichner für die momentan auszugebende Zelle zurück. Die erste Zeile hat den Index 0.

**Parameter:**

**Zahl** (optional) Zeilen-Ebene (0= unterste Ebene bzw. innerste Gruppierung, 1= nächsthöhere, ...). Voreinstellung: 0.

**Rückgabewert:**

Zeichenkette

---

**Crosstab.Row**

**Aufgabe:**

Gibt (in Kreuztabellen) den Zeilenindex für die momentan auszugebende Zelle zurück. Wird diese Funktion im Zeilenbezeichner verwendet bzw. ausgegeben, wird immer Null zurückgegeben, da zum Zeitpunkt der Bestimmung des Label-Textes nur die Koordinate, aber nicht der Spalten-/Zeilenindex bekannt ist.

**Parameter:**

**Boolean** (optional) True: Nur Zellen der gleichen Ebene zählen (Voreinstellung False).

**Rückgabewert:**

Zahl

---

**Crosstab.Value**

---

**Aufgabe:**

Gibt (in Kreuztabellen) den Inhalt der Zelle zurück.

**Parameter: -****Rückgabewert:**

Zahl

---

**CStr\$**

---

**Aufgabe:**

Formatiert eine Zahl mit Hilfe einer Format-Zeichenkette. Diese ist identisch mit der Formatierungsinformation für die printf()-Funktion der Sprache C. Hierbei wird der erste Parameter als numerischer Wert doppelter Präzision übergeben, der Konversionsoperator kann z.B. folgende Werte annehmen:

'f', 'g', 'G', 'e', 'E'.

**Parameter:**

**Zahl**

**Zeichenkette** Formatierung in C-Notation, also meist "%<Formatierung>f".

**Rückgabewert:**

**Zeichenkette**

**Beispiel:**

CStr\$ (1234.5678, "%5.2f") Ergebnis: "1234.78"

CStr\$ (Pi, "%5.1f") Ergebnis: "3.1"

CStr\$ (100\*Pi, "nun: %g") Ergebnis: nun: "3.141593e+02"

---

**Date**

---

**Aufgabe:**

Wandelt eine Zeichenkette in ein Datum um.

- Wenn die Zeichenkette einen Punkt "." enthält, wird er als "t.m.j" eingelesen (deutsch).
- Wenn er einen Schrägstrich "/" enthält, wird er als "m/t/j" eingelesen (englisch).
- Wenn er einen Strich "-" enthält, wird er als "j-m-t" eingelesen (ANSI).
- Wenn er nicht fehlerlos interpretiert werden kann, repräsentiert das Datum einen Wert, der größer ist als alle anderen Daten (1e100), das Rückgabedatum kann also über "<JulianToDate(1e100)" auf Korrektheit geprüft werden.

**Parameter:**

**Zeichenkette**

**Rückgabewert:**

**Datum**

**Beispiel:**

Date ("04.07.1776")	Ergebnis: 04.07.1776
Date ("07/04/1776")	Ergebnis: 07/04/1776
Date ("1776-07-04")	Ergebnis: 1776-07-04

**Hinweis:**

Werden ein- oder zweistellige Jahreszahlen übergeben, so werden alle Werte unter 30 dem 21. Jahrhundert (also 20xx) und alle darüber dem 20. Jahrhundert (also 19xx) zugeordnet.

## **Date\$**

---

**Aufgabe:**

Wandelt ein Datum über eine Format-Zeichenkette in eine entsprechend formatierte Zeichenkette um.

Aufbau der Format-Zeichenkette: Dies ist eine gewöhnliche Zeichenkette, in dem Platzhalter eingebettet werden können.

Platzhalter	Bedeutung
%d	Tag (1..31)
%<n>d	Tag auf <n> Stellen
%0<n>d	Tag auf <n> Stellen, links mit '0'en aufgefüllt
%w	Wochentag (1..7)
%<n>w	Wochentag auf <n> Stellen

%0<n>w	Wochentag auf <n> Stellen, links mit '0'en aufgefüllt
%m	Monat (1..12)
%<n>m	Monat auf <n> Stellen
%0<n>m	Monat auf <n> Stellen, links mit '0'en aufgefüllt
%y	Jahr
%<n>y	Jahr, auf <n> Stellen
%0<n>y	Jahr, auf <n> Stellen, links mit '0'en aufgefüllt
%D	Wochentag, ausgeschrieben
%M	Monat, ausgeschrieben
%Q	Quartal
%"%e", "%<n>e"	Jahr im lokalisierten Kalender (Japan: Emperor's year)
%"%g", "%<n>g"	Area des lokalisierten Kalenders (Japan: Emperor area)
%"%g", "%1g"	Einbuchstabil, lateinischer Buchstabe
%"%gg", "%2g"	Einbuchstabil, lokalisiert
%"%ggg", "%3g"	Langname, lokalisiert
%"%gggg", "%4g"	Langname, lateinische Buchstaben
%"%x"	Lokalisiertes Datum, Kurzform
%H	Stunden im 24-Stundenformat
%h	Stunden im 12-Stundenformat
%<n>h	Stunden auf <n> Stellen
%i	Minuten
%<n>i	Minuten auf <n> Stellen
%s	Sekunden
%<n>s	Sekunden auf <n> Stellen
%PM	Anzeige der Tageshälfte (AM oder PM)

Sofern eine der obigen Formatierungsanweisungen verwendet wird, kann über den optionalen dritten Parameter das zu verwendende Locale festgelegt werden. Enthält der zweite Parameter hingegen ein gültiges ISO 3166-Länderkürzel, so kann über den dritten Parameter angegeben werden, ob das kurze ("0") oder lange ("1") Datumsformat zurückgegeben werden soll.

**Parameter:**

**Datum**                      Zu formatierender Wert.

**Zeichenkette** (optional) Formatbeschreibung oder Landeskürzel.

**Zeichenkette** (optional) ISO 3166-Landeskürzel oder Datumsformat .

**Rückgabewert:**

**Zeichenkette**

**Beispiel:**

Date\$(Date("04.07.1776"),"USA","1") Ergebnis:"Thursday, July 4, 1776"

Date\$(Date("04.07.1776"),"DEU","1") Ergebnis: "Donnerstag, 4. Juli 1776"

Date\$(Date("04.07.1776"),"%2w.Woche; %2d.%2m.%4y")

Ergebnis: "4. Woche; 4. 7. 1776"

Date\$(Date("04.07.1776"),"%02d.%02m.%4y") Ergebnis: "04.07.1776"

Date\$(Now(),"%02H:%02i:%02s") Ergebnis: "16:12:40"

---

## DateDiff

**Aufgabe:**

Berechnet den Abstand der übergebenen Datumswerte in Tagen.

**Parameter:**

**Datum** erster Datumswert

**Datum** zweiter Datumswert

**Rückgabewert:**

**Zahl**

**Beispiel:**

DateDiff (Date("04.07.1776"), Date("04.04.1975")) Ergebnis: 72591

---

## DateDiff\$

**Aufgabe:**

Berechnet den Abstand der übergebenen Datumswerte in Tagen und gibt ihn gemäß der Formatierung als Zeichenkette zurück.

**Parameter:**

**Datum** erster Datumswert

**Datum** zweiter Datumswert

**Zeichenkette** (optional) Formatierung

**Rückgabewert:**

**Zeichenkette**

**Beispiel:**

DateDiff\$(Date("04.07.1776"), Date("04.04.1975")) Ergebnis: "198 Jahre, 9 Monate"

## DateHMS

---

**Aufgabe:**

Wandelt drei Zahlen für Stunde, Minute und Sekunde in ein Datum um.

**Parameter:**

Zahl	Stunde
Zahl	Minute
Zahl	Sekunde

**Rückgabewert:**

Datum

**Beispiel:**

DateHMS (55554000,90000000,45)                      Ergebnis: 03.08.1796

## DateInLeapYear

---

**Aufgabe:**

Überprüft, ob das angegebene Datum in einem Schaltjahr liegt oder nicht. Dabei wird die Berechnung nach dem proleptischen gregorianischen Kalender vorgenommen.

**Parameter:**

Datum	Zu überprüfendes Datum.
-------	-------------------------

**Rückgabewert:**

Boolean

**Beispiel:**

DateInLeapYear (Date("04.07.1176"))                      Ergebnis: True

## DateInRange

---

**Aufgabe:**

Testet, ob das Datum sich innerhalb des durch die Grenzen angegebenen Intervalls befindet. Das Intervall ist abgeschlossen, also ist ein mit einer Grenze identischer Wert noch innerhalb. Um halboffene Intervalle zu definieren, kann die die JulianToDate()-Funktion verwendet werden:

Minimales Datum: JulianToDate(0)

Maximales Datum: JulianToDate(1e100)

**Parameter:**

Datum	Zu überprüfendes Datum.
Datum	Untere Grenze des Testintervalls.
Datum	Obere Grenze des Testintervalls.

**Rückgabewert:**

Boolean

**Beispiel:**

DateInRange (Date('04.07.1776'),Date('01.01.0001'),Date('04.04.1975'))  
Ergebnis: True

**DateToJulian**

---

**Aufgabe:**

Bestimmt den julianischen Wert eines Datums. Dabei wird jedem Tag (auch solchen in der Vergangenheit) eine eindeutige Zahl zugeordnet.

**Parameter:**

Datum

**Rückgabewert:**

Zahl

**Beispiel:**

DateToJulian (Date("04.07.1776"))                      Ergebnis: 2369915

**DateYMD**

---

**Aufgabe:**

Wandelt drei Zahlen für Tag, Monat und Jahr in ein Datum um.

**Parameter:**

Zahl	Jahr
Zahl	Monat
Zahl	Tag

**Rückgabewert:**

Datum

**Beispiel:**

DateYMD (1776,7,4)                                      Ergebnis: 04.07.1776

**Day**

---

**Aufgabe:**

Bestimmt den Tag (1..31) des Monats eines Datums und gibt ihn als Zahl zurück.

**Parameter:**

Datum

**Rückgabewert:**

**Zahl**

**Beispiel:**

Day (Date("04.07.1776"))

Ergebnis: 4

---

## Day\$

**Aufgabe:**

Bestimmt den Tag (1..31) des Monats eines Datums und gibt ihn als Zeichenkette zurück.

**Parameter:**

**Datum**

**Rückgabewert:**

**Zeichenkette**

**Beispiel:**

Day\$ (Date("04.07.1776"))

Ergebnis: "4"

---

## Decade

**Aufgabe:**

Bestimmt das Jahrzehnt eines Datums und gibt es als Zahl zurück. Dieser Wert ist immer relativ zum Jahrhundert (mögliche Werte: 1-10).

**Parameter:**

**Datum**

**Boolean** (optional) Bestimmt, ob die Berechnung "einfach" (Jahrzehnt beginnt mit Jahr 0) oder historisch (Jahrzehnt beginnt mit Jahr 1) durchgeführt werden soll. Voreinstellung: False (historisch).

**Rückgabewert:**

**Zahl**

**Beispiel:**

Decade (Date("04.07.1776"))

Ergebnis: 7

Decade (Date("04.07.1776"),True)

Ergebnis: 8

---

## Distinct

**Aufgabe:**

Wirkt auf die übergeordnete Aggregatsfunktion (z.B. Sum(), Avg(), Count()...) und bewirkt, dass gleiche Werte nur einfach in die Berechnung eingehen.

**Parameter:**

**Alle**

**Rückgabewert:**

Alle

**Beispiel:**

CountIf (Distinct(Customers.Region="D"))

NthLargest (Artikel.Stkpreis,Count (Distinct (Artikel.Stkpreis),True) -1, True)

berechnet den zweitkleinsten Wert und berücksichtigt dabei mehrfach vorkommende Werte nur einmal.

---

**Dow**

**Aufgabe:**

Gibt den Wochentag als Zahl zurück (1..7), 1=Sonntag, 2=Montag, ...

**Parameter:**

Datum

**Rückgabewert:**

Zahl

**Beispiel:**

Dow (Date("04.07.1776"))

Ergebnis: 4 (Donnerstag)

---

**Dow\$**

**Aufgabe:**

Gibt den Wochentag als Zeichenkette gemäß der Ländereinstellung zurück, "Sonntag", "Montag", ...

**Parameter:**

Datum

**Rückgabewert:**

Zeichenkette

**Beispiel:**

Dow\$ (Date("04.07.1776"))

Ergebnis: "Donnerstag"

---

**Drawing**

**Aufgabe:**

Wertet Zeichenkette als Pfad für eine Bilddatei aus.

**Parameter:**

Zeichenkette

**Rückgabewert:**

Bild

## **Drawing\$**

---

**Aufgabe:**

Gibt Bild-Pfad als Zeichenkette zurück.

**Parameter:**

Bild

**Rückgabewert:**

Zeichenkette

## **DrawingHeightSCM**

---

**Aufgabe:**

Gibt die Höhe des Bildes in SCM-Einheiten (1/1000 mm) zurück.

**Parameter:**

Bild

**Rückgabewert:**

Zahl

## **DrawingWidthSCM**

---

**Aufgabe:**

Gibt die Breite des Bildes in SCM-Einheiten (1/1000 mm) zurück.

**Parameter:**

Bild

**Rückgabewert:**

Zahl

## **Empty**

---

**Aufgabe:**

Testet, ob die Zeichenkette leer ist. Wenn der zweite Parameter True ist, wird die Zeichenkette noch von den umgebenden Leerzeichen entfernt (implizites Atrim\$ ()).

**Parameter:**

Zeichenkette

Boolean (optional)

**Rückgabewert:**

Boolean

**Beispiele:**

Empty ("Hallo Welt")

Ergebnis: False

Empty ("")

Ergebnis: True

## EndsWith

---

### Aufgabe:

Testet, ob die Zeichenkette im ersten Argument mit der Zeichenkette im zweiten Argument endet.

### Parameter:

**Zeichenkette**

**Zeichenkette**

**Boolean**

Gibt an, ob Groß-/Kleinschreibung beachtet werden soll.  
Voreinstellung: False

### Rückgabewert:

Boolean

### Beispiele:

EndsWith ("Hallo Welt", "elt")

Ergebnis: True

EndsWith ("Hallo Welt", "llo")

Ergebnis: False

## Evaluate

---

### Aufgabe:

Berechnet die übergebene Formel.

### Parameter:

**Zeichenkette**

### Rückgabewert:

Alle

### Beispiel:

Evaluate("5\*5")

Ergebnis: 25

Str\$(Evaluate("4-3"),0,0)

Ergebnis: 1

## Even

---

### Aufgabe:

Testet, ob eine Zahl gerade ist oder nicht. Im geraden Fall wird True zurückgegeben, sonst False.

### Parameter:

**Zahl**

**Rückgabewert:**

Boolean

**Beispiel:**

Even(2)

Ergebnis: True

## **Exists**

---

**Aufgabe:**

Prüft, ob die Variable bzw. das Feld mit dem übergebenen Namen existiert.  
Wird häufig im Zusammenhang mit GetValue() verwendet.

**Parameter:**

Zeichenkette

**Rückgabewert:**

Boolean

**Beispiel:**

Exists("CustomerID")

Ergebnis: False

If(Exists("Kunde.Status"), Evaluate("Kunde.Status"), "In Ihrer Datenbank ist kein Status für den Kunden vergeben")

## **Exp**

---

**Aufgabe:**

Gibt den Exponenten zur Basis e zurück ( $e^x$ ).

**Parameter:**

Zahl

**Rückgabewert:**

Zahl

**Beispiel:**

Exp(3)

Ergebnis: 20.08553692

## **Exp10**

---

**Aufgabe:**

Gibt den 10er Exponenten zurück ( $10^x$ ).

**Parameter:**

Zahl

**Rückgabewert:**

Zahl

**Beispiel:**

Exp10(3)

Ergebnis: 1000

## FirstHeaderThisTable

---

### Aufgabe:

Liefert zurück, ob die Kopfzeile der aktuellen Tabelle zum ersten Mal ausgegeben wird. Diese Funktion kann als Darstellungsbedingung der Kopfzeile verwendet werden, um die erneute Ausgabe der Kopfzeile zu unterdrücken, falls der Druck einer Tabelle aus Platzmangel auf der Folgeseite fortgesetzt wird. Die Kopfzeile wird dann nur zu Beginn der Tabelle ausgegeben. Diese Funktion steht nur im Multitabellenmodus zur Verfügung.

Parameter: -

Rückgabewert:

Boolean

## Floor

---

### Aufgabe:

Berechnet ausgehend vom übergebenen Wert die nächstkleinere Ganzzahl. Siehe auch Funktion Ceil().

Parameter:

Zahl

Rückgabewert:

Zahl

Beispiel:

Floor (5.6)

Ergebnis: 5

## Frac

---

### Aufgabe:

Gibt den Nachkommaanteil einer Zahl zurück

Parameter:

Zahl

Rückgabewert:

Zahl

Beispiel:

Frac (1.234)

Ergebnis: 0.234

Frac (Pi)

Ergebnis: 0.1415926535

## Fstr\$

### Aufgabe:

Formatiert eine Zahl mit Hilfe der Format-Zeichenkette.

Dieser besteht aus folgenden Zeichen ('wenn negativ' bezieht sich auf den zu formatierenden Wert):

*	Ziffer oder '*'-Präfix
\$	Lokales Währungszeichen
-	Ziffer oder Vorzeichen, wenn negativ
+	Ziffer oder Vorzeichen
(	Ziffer oder '('-Präfix wenn negativ
)	')'-Postfix wenn negativ
#	Ziffer oder Leerzeichen-Präfix
&	Ziffer oder '0'
.	Dezimalpunkt
,	Komma, oder Leerzeichen-Präfix

Ein Präfix ist dabei ein Zeichen, das einem Wert bei Bedarf vorangestellt wird. Der Ausdruck `Fstr$(1, "***")` ergibt z.B. `"**1"`. Dem Wert `"1"` werden dabei die Zeichen `"**"` vorangestellt, weil es kürzer als die drei reservierten Stellen ist. Ein Postfix ist ein Zeichen, das einem Wert bei Bedarf nachgestellt wird. Diese Formatierungs-Zeichen dürfen beliebig kombiniert werden. Wenn die Zahl zu groß ist für die gewünschte Formatierung, wird eine Zeichenkette aus `'*'` erstellt.

Über den dritten (optionalen) Parameter kann zusätzlich die Formatierung gesteuert werden.

Wert	Bedeutung
1	Entfernung führender Leerzeichen. Die Anwendung entspricht den Funktionen <code>RTrim\$()</code> und <code>LTrim\$()</code> .
2	Leere Zeichenkette wenn Wert Null.
3	Entfernung führender Leerzeichen und leere Zeichenkette wenn Wert Null.
4	Entfernung von '0' am Ende der Dezimalstellen.

### Parameter:

**Zahl**

**Zeichenkette**    Format-Zeichenkette

Zahl	(optional) zusätzliche Formatierung
<b>Rückgabewert:</b>	
<b>Zeichenkette</b>	
<b>Beispiele:</b>	
FStr\$(3.142,"#")	Ergebnis: "3"
FStr\$(0,"#",2)	Ergebnis: ""
FStr\$(0003.142,"#.###",1)	Ergebnis: "3,142"
Fstr\$(3.1,"#####&.#",4)	Ergebnis: "3,1"
Fstr\$(3.1,"#####&.#")	Ergebnis: "3,100"
Fstr\$(.1,"#####&.#")	Ergebnis: "0,100"
FStr\$(5003.1,"#,##&.#")	Ergebnis: "5.003,10"
FStr\$(3.142,"#.###")	Ergebnis: "3,142"
FStr\$(3.142,"#####")	Ergebnis: "*****"
FStr\$(3.142,"(###)")	Ergebnis: " 3,142 "
FStr\$(-3.142,"(###)")	Ergebnis: "(3,142)"
FStr\$(3.142,"+#.###")	Ergebnis: "+3,142"
FStr\$(3.142,"-#.###")	Ergebnis: " 3,142"
FStr\$(-3.142,"-#.###")	Ergebnis: "-3,142"
FStr\$(3.142,"&&&.&&&")	Ergebnis: "003,142"
FStr\$(3.142,"***.***")	Ergebnis: "***3,142"
FStr\$(3.142,"\$\$\$.\$\$\$")	Ergebnis: "\$\$3,142"
FStr\$(3.142,"###.***")	Ergebnis: " 3,142"
FStr\$(5003.1,"#,##&.#")	Ergebnis: "5.003,10"
FStr\$(3.142,"#####")	Ergebnis: " 3"

## GeometricAvg

---

### Aufgabe:

Bestimmt das geometrische Mittel. Das geometrische Mittel ist ein geeignetes Lagemaß für Größen, von denen das Produkt anstelle der Summe interpretierbar ist, z.B. von Verhältnissen oder Wachstumsraten.

### Parameter:

<b>Zahl</b>	Ausdruck für den zu mittelnden Wert
<b>Boolean</b>	(optional) True: nach der Ausgabe werden die für die Berechnung gemerkten Werte gelöscht (Voreinstellung: True). Bitte beachten Sie, dass die für die Berechnung gemerkten Werte generell bei jedem (Unter-) Tabellenende gelöscht werden. Der zweite Parameter bestimmt lediglich,

ob die Werte bei einer Ausgabe schon innerhalb der Tabelle gelöscht werden.

**Rückgabewert:**

Zahl

## **GetValue**

---

**Aufgabe:**

Liefert den Wert der Variablen bzw. des Feldes mit dem übergebenen Namen. Wird häufig im Zusammenhang mit Exists() verwendet.

**Parameter:**

Zeichenkette

**Rückgabewert:**

Alle

**Beispiel:**

Str\$(GetValue("Customers.CustomerID"),0,0)      Ergebnis: 1234

If (Exists("Kunde.Status"), Evaluate("Kunde.Status"), "In Ihrer Datenbank ist kein Status für den Kunden vergeben")

## **GetVar**

---

**Aufgabe:**

Holt einen Wert, der über die SetVar()-Funktion gesetzt wurde, aus dem Variablenspeicher. Die Aufgabe dieser Funktionen ist die eines einfachen Zwischenspeichers für Werte. Sie sollten keine komplexen Verschachtelungen mit GetVar/SetVar durchführen oder die beiden Funktionen miteinander kombinieren - hier können, insbesondere in Kopf-, Fuß- und Gruppenzeilen, unerwartete Effekte auftreten.

**Parameter:**

Alle

**Rückgabewert:**

Alle

**Beispiel:**

GetVar ("Seite")

Ergebnis: Inhalt von SetVar

## **Hour**

---

**Aufgabe:**

Bestimmt die Stunde des Datums und gibt sie als Zahl zurück. Wenn der Parameter weggelassen wird, wird die Stunde der Druckzeit zurückgeliefert.

**Parameter:**

Datum (optional)

**Rückgabewert:**

Zahl

**Beispiel:**

Hour()=10

## HSL

---

**Aufgabe:**

Berechnet einen Farbwert im HSL-Farbraum, bei dem man die Farbe mit Hilfe des Farbtons (englisch hue), der Farbsättigung (saturation) und der relativen Helligkeit (lightness) bestimmt.

**Parameter:**

**Zahl** Farbton [0-360] als Farbwinkel H auf dem Farbkreis (z.B. 0°=Rot, 120°=Grün, 240°=Blau).

**Zahl** Sättigung [0-1] als Intervall von Null bis Eins (0=Neutralgrau, 0.5= wenig gesättigte Farbe, 1=gesättigte, reine Farbe).

**Zahl** Helligkeit bzw. Dunkelstufe [0-1] als Intervall von Null bis Eins (0=keine Helligkeit, 1=volle Helligkeit).

**Rückgabewert:**

Zahl

## Hyperlink\$

---

**Aufgabe:**

Die Funktion Hyperlink\$ erzeugt für den HTML und MHTML-Export einen Hyperlink.

Wenn vorhanden, wird der Hyperlink nur eingebettet, wenn der 3. Parameter True (=Voreinstellung) ergibt.

Enthält ein Objekttext die Zeichenfolge:

```
<!--beginn:hyperlink="ziel"-->"Anzeigetext"<!--end:hyperlink-->
```

dann wird an dieser Stelle automatisch ein Hyperlink auf das angegebene Ziel erzeugt. Die Hyperlink-Funktion erstellt automatisch eine Zeichenkette mit der benötigten Syntax.

**Parameter:**

**Zeichenkette** Anzeigetext

**Zeichenkette** Hyperlink

**Boolean** (optional) Einbetten

Rückgabewert:

Zeichenkette

Beispiel:

Hyperlink\$ ("domain.com",<http://www.domain.com>)

---

## If

vgl. Cond()

---

## IssueIndex

Aufgabe:

Gibt den Index der Ausfertigung (1..) zurück. Verfügbar in einer Anzeige- und Layoutbereichsbedingung wenn in den Projektparametern mehrere Ausfertigungen gewählt wurden.

Parameter:

Zahl

Rückgabewert:

Zahl

---

## Int

Aufgabe:

Gibt den ganzzahligen Anteil einer Zahl zurück. Der Wert wird abgeschnitten.

Parameter:

Zahl

Rückgabewert:

Zahl

Beispiele

Int(3,1)

Ergebnis: 3

---

## IsNull

Aufgabe:

Überprüft ob der übergebene Wert oder das Ergebnis des Ausdrucks Null ist, also ein leerer Feldinhalt.

Parameter:

Alle

Rückgabewert:

Boolean

## IsNullOrEmpty

---

### Aufgabe:

Überprüft eine Zeichenkette, ob sie leer oder Null() ist.

### Parameter:

**String** die zu überprüfende Zeichenkette  
**Boolean** (optional) Wenn der Wert True ist, dann werden Leerzeichen am Anfang und Ende der Zeichenkette entfernt. Voreinstellung: False

### Rückgabewert:

Zeichenkette

### Beispiel:

IsNullOrEmpty (" ", True) Ergebnis: True

## Join\$

---

### Aufgabe:

Gibt die gesammelten Zeichenketten zurück, getrennt durch ein Trennzeichen.

### Parameter:

**Zeichenkette** Werte  
**Zeichenkette** (optional) Trennzeichen für die Werte (Voreinstellung: ";").  
**Zahl** (optional) Maximalzahl der Werte (danach wird '...' angehängt). Voreinstellung: alle Werte".  
**Boolean** (optional) True: nach der Ausgabe werden die gemerkten Werte gelöscht (Voreinstellung: True). Bitte beachten Sie, dass die gemerkten Werte generell bei jedem (Unter-)Tabellenende gelöscht werden. Der zweite Parameter bestimmt lediglich, ob die Werte bei einer Ausgabe schon innerhalb der Tabelle gelöscht werden.

### Rückgabewert:

Zeichenkette

### Beispiel:

Join\$(Str\$(Anzahl \* Stückpreis,0,2),"+") Ergebnis: "12,55 + 33,45 + 12,12"

## JulianToDate

---

### Aufgabe:

Interpretiert eine Zahl als julianisches Datum (jedem Tag wird eine eindeutige Zahl zugeordnet) und gibt das entsprechende Datum zurück.

### Parameter:

**Zahl****Rückgabewert:****Datum****Beispiel:**

JulianToDate(2369915)

Ergebnis: 04.07.1776

## LangCase\$

---

**Aufgabe:**

Gibt entsprechend der eingestellten Sprache einen der Teile der Zeichenkette zurück.

**Parameter:**

**Zeichenkette** Zeichenkette die zurückgegeben wird, wenn keine Lokalisierungszeichenkette gefunden werden kann. Hierbei muss es sich um ein gültiges ISO 639-Sprachkürzel handeln.

**Zeichenkette** Zeichenkette mit Übersetzungstexten, getrennt durch "|" (oder durch optionalen dritten Parameter). Übersetzungstexte müssen in "ISO 639-Sprachkürzel = Übersetzungstext |[ISO 639-Sprachkürzel = Übersetzungstext |...] formatiert sein. Um "|" oder "=" im Wert oder Schlüssel verwenden zu können, stellen Sie diesen ein "\" voran, z.B. "USA=He\=llo".

Wir empfehlen die Verwendung eines vollständigen Sprachkürzels im Format <ISO639>-<ISO3166> wie z.B. de-de oder en-us.

<http://msdn.microsoft.com/en-us/library/0h88fahh%28VS.85%29.aspx>

**Zeichenkette** (optional) Trennzeichen (Voreinstellung: "|")

**Rückgabewert:****Zeichenkette****Beispiel:**

LangCase\$("Hallo","USA=Hello|ESP=Hóla") Ergebnis: "Hallo" (auf dt. System)

## LastFooterThisTable

---

**Aufgabe:**

Liefert zurück, ob die Fußzeile der aktuellen Tabelle zum letzten Mal ausgegeben wird. Diese Funktion kann als Darstellungsbedingung der Fußzeile verwendet werden, um die Ausgabe der Fußzeile zu unterdrücken, falls der Druck einer Tabelle aus Platzmangel auf der Folgeseite fortgesetzt wird. Die Fußzeile wird dann nur auf der letzten Seite der Tabelle ausgegeben. Diese Funktion steht nur im Berichtscontainer zur Verfügung.

**Parameter:**

-

**Rückgabewert:**

Boolean

## **Lastpage**

---

**Aufgabe:**

Gibt an, ob die momentane Seite die letzte Seite ist. Diese Funktion kann nur in Fußzeilen von Tabellen, als Bedingung in Layoutbereichen oder in an Tabellen angehängten Objekten (solange nicht während des Drucks der angehängten Objekte ein Seitenumbruch erfolgt ist) verwendet werden! In allen anderen Fällen ist Lastpage() immer False.

**Parameter:**

-

**Rückgabewert:**

Boolean

**Beispiel:**

Cond>Lastpage(),"Endsumme","Zwischensumme")

## **Left\$**

---

**Aufgabe:**

Kürzt eine Zeichenkette von rechts um so viele Zeichen, dass gerade noch die unter Zahl angegebene Anzahl von Zeichen übrigbleibt. Wenn die ursprüngliche Zeichenkette schon klein genug ist, wird diese nicht verändert.

**Parameter:**

<b>Zeichenkette</b>	Der zu kürzende Wert.
<b>Zahl</b>	max. Anzahl der Stellen des Resultats (inkl. Punkte).
<b>Boolean</b>	(optional) True: Der abgeschnittene Wert wird mit "..." beendet (Voreinstellung False). Bei Zahl < 3 wird die Einstellung ignoriert.

**Rückgabewert:**

Zeichenkette

**Beispiele:**

Left\$ ("ABCD", 2)                      Ergebnis: "AB"

## **Len**

---

**Aufgabe:**

Gibt die Anzahl der Zeichen einer Zeichenkette zurück.

**Parameter:**

**Zeichenkette**

**Rückgabewert:**

**Zahl**

**Beispiel:**

Len("1234"+"56"+"78")                      Ergebnis: 8

---

## LoadFile\$

**Aufgabe:**

Gibt den Inhalt der Datei als Zeichenkette aus.

Wichtig: Bitte beachten Sie, dass die Darstellung von RTF-Texten aus Variablen/Feldern auf Inhalte ausgelegt ist, welche mit Hilfe des Microsoft RTF-Ccontrols erzeugt wurden. Sie können diese Inhalte beispielsweise mit der Windows Anwendung "Wordpad" generieren. Inhalte, die in Microsoft Word erzeugt wurden, sind unter Umständen nicht mit dem vom Control verwendeten RTF-Standard kompatibel und sollten deshalb auch nicht verwendet werden.

**Parameter:**

**Zeichenkette**

**Zeichenkette**                      (optional) Wenn die Datei nicht vorhanden ist, wird der hier angegebene Wert verwendet.

**Rückgabewert:**

**Zeichenkette**

**Beispiel:**

LoadFile\$(ProjectPath\$()+"\agb.txt","Datei ist nicht vorhanden")

---

## Locale\$

**Aufgabe:**

Gibt Informationen zu den Ländereinstellungen zurück wie z.B. Währung, Dezimalstellen, Separatoren, Sprache, Länderkürzel. Das Kürzel für das entsprechende Land wird im zweiten Parameter angegeben - wenn kein zweiter Parameter angegeben ist, wird das eingestellte Land verwendet.

**Parameter:**

**Zahl**                                      Index des Locale-Eintrags.

<http://msdn.microsoft.com/en-us/library/bb507201.aspx>

**Zeichenkette** (optional) ISO-Kürzel der Sprache und des Landes, dessen Formatierung verwendet werden soll. Format: <ISO639>-<ISO3166> wie z.B. de-de oder en-us.

<http://msdn.microsoft.com/en-us/library/0h88fahh%28VS.85%29.aspx>

**Rückgabewert:**

**Zeichenkette**

**Beispiel:**

Locale\$ (42,"de-de")

Ergebnis: "Montag"

---

## LocCurr\$

**Aufgabe**

Gibt eine Zeichenkette mit der für das Land gültigen Währungsformatierung ohne Währungssymbol zurück.

**Parameter:**

**Zahl** Zu formatierender Wert.

**Zeichenkette** (optional) ISO-Kürzel der Sprache und des Landes, dessen Formatierung verwendet werden soll. Format: <ISO639>-<ISO3166> wie z.B. de-de oder en-us.

Voreinstellung: eingestellte Locale

<http://msdn.microsoft.com/en-us/library/0h88fahh%28VS.85%29.aspx>

**Rückgabewert:**

**Zeichenkette**

**Beispiel:**

LocCurr\$ (50,"de-de")

Ergebnis: "50,00"

---

## LocCurrL\$

**Aufgabe**

Gibt eine Zeichenkette mit der für das Land gültigen Währungsformatierung mit Währungssymbol zurück.

**Parameter:**

**Zahl** Zu formatierender Wert.

**Zeichenkette** (optional) ISO-Kürzel der Sprache und des Landes, dessen Formatierung verwendet werden soll. Format: <ISO639>-<ISO3166> wie z.B. de-de oder en-us.

Voreinstellung: eingestellte Locale

<http://msdn.microsoft.com/en-us/library/0h88fahh%28VS.85%29.aspx>

**Rückgabewert:**

**Zeichenkette**

**Beispiel:**

LocCurr\$ (50,"de-de")

Ergebnis: "50,00 EUR"

**LocDate\$****Aufgabe**

Gibt eine Zeichenkette mit dem für das Land gültigen Datumsformat zurück.

**Parameter:**

<b>Datum</b>	Datum das formatiert werden soll.
<b>Zeichenkette</b>	(optional) Ländercode/ISO-Kürzel der Sprache und des Landes, dessen Formatierung verwendet werden soll. Format: <ISO639>-<ISO3166> wie z.B. de-de oder en-us. Voreinstellung: eingestellte Locale <a href="http://msdn.microsoft.com/en-us/library/0h88fahh%28VS.85%29.aspx">http://msdn.microsoft.com/en-us/library/0h88fahh%28VS.85%29.aspx</a>
<b>Zahl</b>	(optional) 0=Kurzform (Voreinstellung), 1=Langform, 2=Kurzform, alternativer Kalender, 3=Langform, alternativer Kalender

**Rückgabewert:**

Zeichenkette

**Beispiel:**

LocDate\$ (Date("04.07.1776"),"de-de")

Ergebnis: "04.07.1776"

LocDate\$(Date("04.07.1776"),"en-us",2)

Ergebnis: "7/4/1776"

**LocDateTime****Aufgabe**

Wandelt die Zeichenkette in ein Datum (mit Zeit wenn erforderlich). Es wird angenommen, dass die Zeichenkette entsprechend dem angegebenen Länderkürzel formatiert ist.

**Parameter:**

<b>Zeichenkette</b>	Datum
<b>Zeichenkette</b>	(optional) ISO-Kürzel der Sprache und des Landes, dessen Formatierung verwendet werden soll. Format: <ISO639>-<ISO3166> wie z.B. de-de oder en-us. Voreinstellung: eingestellte Locale <a href="http://msdn.microsoft.com/en-us/library/0h88fahh%28VS.85%29.aspx">http://msdn.microsoft.com/en-us/library/0h88fahh%28VS.85%29.aspx</a>

**Rückgabewert:**

Datum

**Beispiel:**

LocDateTime("07/04/1776","en-us")

Ergebnis: 04.07.1776

---

## LocNumber\$

---

### Aufgabe

Gibt eine Zeichenkette mit der für das Land gültigen Zahlenformatierung zurück.

### Parameter:

**Zahl** Zu formatierender Wert.

**Zeichenkette** (optional) ISO-Kürzel der Sprache und des Landes, dessen Formatierung verwendet werden soll. Format: <ISO639>-<ISO3166> wie z.B. de-de oder en-us.

Voreinstellung: eingestellte Locale

<http://msdn.microsoft.com/en-us/library/0h88fahh%28VS.85%29.aspx>

### Rückgabewert:

**Zeichenkette**

### Beispiel:

LocNumber\$ (123,"de-de")

Ergebnis: "123,00"

---

## LocTime\$

---

### Aufgabe

Gibt eine Zeichenkette mit der für das Land gültigen Zeitformatierung zurück.

### Parameter:

**Datum** Zu formatierender Wert.

**Zeichenkette** (optional) ISO-Kürzel der Sprache und des Landes, dessen Formatierung verwendet werden soll. Format: <ISO639>-<ISO3166> wie z.B. de-de oder en-us.

Voreinstellung: eingestellte Locale

<http://msdn.microsoft.com/en-us/library/0h88fahh%28VS.85%29.aspx>

**Zahl** (optional) 0= normal (Voreinstellung), 1=immer 24h-Format, 2=normal, kein AM/PM, 3=24h-Format, kein AM/PM

### Rückgabewert:

**Zeichenkette**

### Beispiel:

LocTime\$ (Now(),"de-de")

Ergebnis: aktuelle Systemzeit

---

## LocVal

---

### Aufgabe

Interpretiert die Zeichenkette als Zahl und gibt deren Wert zurück. Ggf. vorkommende lokalisierte Dezimal- und Tausendertrennzeichen werden berücksichtigt.

**Parameter:**

**Zeichenkette** Zahl (als Zeichenkette).

**Zeichenkette** (optional) ISO-Kürzel der Sprache und des Landes, dessen Formatierung verwendet werden soll. Format: <ISO639>-<ISO3166> wie z.B. de-de oder en-us.

Voreinstellung: eingestellte Locale

<http://msdn.microsoft.com/en-us/library/0h88fahh%28VS.85%29.aspx>

**Rückgabewert:**

Zahl

**Beispiel:**

LocVal ("12","de-de")	Ergebnis: 12,00
LocVal ("12,00","en-us")	Ergebnis: 1200,00

---

## Log

**Aufgabe:**

Gibt den Logarithmus zur Basis e zurück ( $\ln(x)$ ).

**Parameter:**

Zahl

**Rückgabewert:**

Zahl

**Beispiel:**

Log (Exp(1))	Ergebnis: 1
Log (3)	Ergebnis: 1,10

---

## Log10

**Aufgabe:**

Gibt den 10er Logarithmus zurück ( $\log(x)$ ).

**Parameter:**

Zahl

**Rückgabewert:**

Zahl

**Beispiel:**

Log10 (1000)	Ergebnis: 3
--------------	-------------

## Lower\$

---

### Aufgabe:

Wandelt die Zeichen in Kleinbuchstaben um.

### Parameter:

Zeichenkette

### Rückgabewert:

Zeichenkette

### Beispiel:

Lower\$ ("HALLO WELT")

Ergebnis: "hallo welt"

## LTrim\$

---

### Aufgabe:

Entfernt die führenden Leerzeichen am Anfang einer Zeichenkette.

### Parameter:

Zeichenkette

### Rückgabewert:

Zeichenkette

### Beispiel:

LTrim\$ (" Hallo Welt")

Ergebnis: "Hallo Welt"

## Max

---

### Aufgabe:

Vergleicht zwei Werte vom Typ Zahl oder zwei Werte vom Typ Datum und gibt den größeren der beiden Werte zurück.

### Parameter:

Zahl oder Datum

Zahl oder Datum

### Rückgabewert:

Zahl oder Datum

### Beispiel:

Max (100,10)

Ergebnis: 100

## Maximum

---

### Aufgabe:

Liefert den größten Wert der Datenmenge, die sich aus dem ersten Argument ergibt.

**Parameter:**

**Zahl**

**Boolean** (optional) True: nach der Ausgabe werden die für die Berechnung gemerkten Werte gelöscht (Voreinstellung: True). Bitte beachten Sie, dass die für die Berechnung gemerkten Werte generell bei jedem (Unter-)Tabellenende gelöscht werden. Der zweite Parameter bestimmt lediglich, ob die Werte bei einer Ausgabe schon innerhalb der Tabelle gelöscht werden.

**Rückgabewert:**

**Zahl**

**Beispiel:**

Maximum (Anzahl \* Stückpreis)

Ergebnis: Maximum Gesamtpreis

---

## Median

---

**Aufgabe:**

Liefert den Median der Datenmenge, die sich aus dem ersten Argument ergibt.

**Parameter:**

**Zahl**

Ausdruck für den zu mittelnden Wert

**Boolean**

(optional) True: nach der Ausgabe werden die für die Berechnung gemerkten Werte gelöscht (Voreinstellung: True). Bitte beachten Sie, dass die für die Berechnung gemerkten Werte generell bei jedem (Unter-) Tabellenende gelöscht werden. Der zweite Parameter bestimmt lediglich, ob die Werte bei einer Ausgabe schon innerhalb der Tabelle gelöscht werden.

**Rückgabewert:**

**Zahl**

**Beispiel:**

Median (Stückpreis)

Ergebnis: Median Stückpreis

---

## Mid\$

---

**Aufgabe:**

Die resultierende Zeichenkette wird aus dem ursprünglichen gewonnen, indem ab der Anfangsposition die gewünschte Anzahl von Zeichen zurückgegeben wird.

Wenn der 3. Parameter wegfällt, dann wird die Zeichenkette von der Anfangsposition bis zum Ende zurückgegeben.

Das erste Zeichen der Zeichenkette entspricht Position 0.

**Parameter:**

**Zeichenkette**

**Zahl** Anfangsposition

**Zahl** (optional) Anzahl der gewünschten Zeichen.

**Rückgabewert:**

**Zeichenkette**

**Beispiel:**

Mid\$ ("Hallo Welt",3,5)

Ergebnis: "lo We"

## Min

---

**Aufgabe:**

Vergleicht zwei Werte vom Typ Zahl oder zwei Werte vom Typ Datum und gibt den kleineren der beiden Werte zurück.

**Parameter:**

**Zahl oder Datum**

**Zahl oder Datum**

**Rückgabewert:**

**Zahl oder Datum**

**Beispiel:**

Min (100,10)

Ergebnis: 10

## Minimum

---

**Aufgabe:**

Liefert den kleinsten Wert der Datenmenge, die sich aus dem ersten Argument ergibt.

**Parameter:**

**Zahl**

**Boolean** (optional) True: nach der Ausgabe werden die für die Berechnung gemerkten Werte gelöscht (Voreinstellung: True). Bitte beachten Sie, dass die für die Berechnung gemerkten Werte generell bei jedem (Unter-)Tabellenende gelöscht werden. Der zweite Parameter bestimmt lediglich, ob die Werte bei einer Ausgabe schon innerhalb der Tabelle gelöscht werden.

**Rückgabewert:**

### Zahl

#### Beispiel:

Minimum (Anzahl \* Stückpreis)

Ergebnis: Minimum Gesamtpreis

---

### Minute

Bestimmt die Minute des übergebenen Datums und gibt sie als Zahl zurück. Wenn der Parameter weggelassen wird, wird die Minute der Druckzeit zurückgeliefert.

#### Parameter:

Datum (optional)

#### Rückgabewert:

Zahl

---

### Mode

#### Aufgabe:

Liefert den Modus (häufigsten Wert) der Datenmenge, die sich aus dem ersten Argument ergibt.

#### Parameter:

**Zahl** Ausdruck für den zu untersuchenden Wert.

**Boolean** (optional) True: nach der Ausgabe werden die für die Berechnung gemerkten Werte gelöscht (Voreinstellung: True). Bitte beachten Sie, dass die für die Berechnung gemerkten Werte generell bei jedem (Unter-) Tabellenende gelöscht werden. Der zweite Parameter bestimmt lediglich, ob die Werte bei einer Ausgabe schon innerhalb der Tabelle gelöscht werden.

#### Rückgabewert:

Zahl

---

### Month

#### Aufgabe:

Bestimmt den Monat (1..12) eines Datums und gibt ihn als Zahl zurück.

#### Parameter:

Datum

**Boolean** (optional) bestimmt ob die Berechnung relativ zum Jahr (1...12, Voreinstellung: False) oder absolut seit 1.1.0001 (1...) zurückgegeben werden soll

#### Rückgabewert:

## Zahl

### Beispiel:

Month (Date("04.07.1776"), False)

Ergebnis: 7

---

## Month\$

### Aufgabe:

Bestimmt den Monat (1..12) eines Datums und gibt ihn als Zeichenkette zurück.

### Parameter:

Datum

### Rückgabewert:

Zeichenkette

### Beispiel:

Month\$ (Date("04.07.1776"))

Ergebnis: "Juli"

---

## Now

### Aufgabe:

Gibt das aktuelle Datum mit Uhrzeit zurück.

### Parameter: -

### Rückgabewert:

Datum

---

## NthLargest

### Aufgabe:

Liefert den n.-größten Wert der Datenmenge, die sich aus dem ersten Argument ergibt.

### Parameter:

Zahl

Zahl

Boolean

<n>, d.h. der Index für den auszugebenden Wert (1-basiert).

(optional) True: nach der Ausgabe werden die für die Berechnung gemerkten Werte gelöscht (Voreinstellung: True). Bitte beachten Sie, dass die für die Berechnung gemerkten Werte generell bei jedem (Unter-)Tabellenende gelöscht werden. Der zweite Parameter bestimmt lediglich, ob die Werte bei einer Ausgabe schon innerhalb der Tabelle gelöscht werden.

### Rückgabewert:

**Zahl**

**Beispiel:**

NthLargest (Stückpreis,2)

berechnet den zweitgrößten Wert

## **NthLargestIndex**

---

**Aufgabe:**

Liefert den Index des n.-größten Werts der Datenmenge, die sich aus dem ersten Argument ergibt. Wird z.B. für NthValue() benötigt.

**Parameter:**

**Zahl**

**Zahl**

<n>, d.h. der Index für den auszugebenden Wert.

**Boolean**

(optional) True: nach der Ausgabe werden die für die Berechnung gemerkten Werte gelöscht (Voreinstellung: True). Bitte beachten Sie, dass die für die Berechnung gemerkten Werte generell bei jedem (Unter-)Tabellenende gelöscht werden. Der zweite Parameter bestimmt lediglich, ob die Werte bei einer Ausgabe schon innerhalb der Tabelle gelöscht werden.

**Rückgabewert:**

**Zahl**

**Beispiel:**

NthLargestIndex (Produkt.Stückpreis,2)

## **NthValue**

---

**Aufgabe:**

Liefert den Inhalt des n. Wertes der Datenmenge, die sich aus dem ersten Argument ergibt. Wird gerne im Zusammenhang mit der übergeordneten Aggregatsfunktion NthLargestIndex() verwendet.

**Parameter:**

**Alle**

**Zahl**

<n>, d.h. der Index für den auszugebenden Wert, berechnet über NthLargestIndex()

**Boolean**

(optional) True: nach der Ausgabe werden die für die Berechnung gemerkten Werte gelöscht (Voreinstellung: True). Bitte beachten Sie, dass die für die Berechnung gemerkten Werte bei jedem (Unter-)Tabellenende gelöscht werden. Der zweite Parameter bestimmt lediglich, ob die Werte bei einer Ausgabe schon innerhalb der Tabelle gelöscht werden.

**Rückgabewert:**

Alle

**Beispiel:**

NthValue (Produkt.Produktname, NthLargestIndex (Produkt.Stückpreis,2))

Ergebnis: Name des zweitteuersten Produktes

---

**Null**

**Aufgabe:**

Ergibt einen Null-Wert (Wert nicht vorhanden).

**Parameter:**

-

**Rückgabewert:**

Alle

---

**NullSafe**

**Aufgabe:**

Überprüft den Parameter auf Null, und gibt einen Ersatzwert zurück, wenn das der Fall ist, ansonsten das Ergebnis des Ausdrucks.

**Parameter:**

Alle

Alle (optional) Wert, der zurückgegeben werden soll, wenn der Ausdruck Null ist (Voreinstellung: Leere Zeichenkette bzw. 0, je nach Typ).

**Rückgabewert:**

Alle

---

**NumInRange**

**Aufgabe:**

Testet, ob die Zahl sich innerhalb des durch die Grenzen angegebenen Intervalls befindet. Das Intervall ist abgeschlossen, also ist ein mit einer Grenze identischer Wert noch innerhalb.

**Parameter:**

Zahl

Zahl Untere Grenze.

Zahl Obere Grenze.

**Rückgabewert:**

### Boolean

#### Beispiel:

NumInRange (50,25,100)    Ergebnis: True

---

### Odd

#### Aufgabe:

Testet, ob die Zahl ungerade ist oder nicht. Im ungeraden Fall wird True zurückgegeben, sonst False.

#### Parameter:

Zahl

#### Rückgabewert:

Boolean

#### Beispiel:

Odd (5)    Ergebnis: True

---

### Ord

#### Aufgabe:

Liefert den ASCII-Wert des ersten Zeichens zurück.

#### Parameter:

Zeichenkette

#### Rückgabewert:

Zahl

#### Beispiel:

Ord("Hallo Welt")    Ergebnis: 72

---

### Page

#### Aufgabe:

Gibt die momentane Seitenzahl zurück.

#### Parameter:

Zahl    (optional) einzuberechnender Offset, wird zur Seitenzahl hinzuaddiert

#### Rückgabewert:

Zahl

#### Beispiel:

Page ()    Ergebnis: aktuelle Seitenzahl

Page (3)    Ergebnis: aktuelle Seitenzahl + 3

## Page\$

---

### Aufgabe:

Gibt die Seitenzahl der Druckseite als Zeichenkette zurück.

### Parameter:

**Zahl** (optional) einzuberechnender Offset, wird zur Seitenzahl hinzuaddiert

### Rückgabewert:

**Zeichenkette**

### Beispiel:

Page\$ () Ergebnis: aktuelle Seitenzahl  
Page\$ (3) Ergebnis: aktuelle Seitenzahl + 3

## Pow

---

### Aufgabe:

Entspricht der Funktion  $(\text{Basis}) ^ (\text{Exponent})$ .

### Parameter:

**Zahl** Basis  
**Zahl** Exponent

### Rückgabewert:

**Zahl**

### Beispiel:

Pow (2,3) Ergebnis: 8

## Previous

---

### Aufgabe:

Gibt den Wert der Variablen, des Feldes bzw. der Formel zurück, den sie bei dem letzten Datensatz hatte.

### Parameter:

**Alle** Variable, Feld oder Formel

### Rückgabewert:

**Alle**

### Beispiel:

Previous (NAME) Ergebnis: vorheriger Name

## PreviousUsed

---

### Aufgabe:

Gibt den Wert der Variablen, des Feldes bzw. der Formel zurück, den sie bei der letzten tatsächlichen Berechnung der Funktions-Instanz hatte.

### Parameter:

**Alle** Variable, Feld oder Formel

### Rückgabewert:

**Alle**

### Beispiel:

PreviousUsed (NAME) Ergebnis: letzter Name

## ProjectParameter\$

---

### Aufgabe:

Gibt den Wert eines Projekt-Parameters aus. Mögliche Parameter:

LL.FAX.Queue	Druck-Warteschlange
LL.FAX.RecipName	Empfängername
LL.FAX.RecipNumber	Empfängerfaxnummer
LL.FAX.SenderName	Absendername
LL.FAX.SenderCompany	Absenderfirma
LL.FAX.SenderDepartment	Absenderabteilung
LL.FAX.SenderBillingCode	Absenderverrechnungscode
LL.MAIL.To	An (To)
LL.MAIL.CC	An (CC)
LL.MAIL.BCC	An (BCC)
LL.MAIL.Subject	Betreff
LL.MAIL.From	Von (From)
LL.MAIL.ReplyTo	Von (ReplyTo)
LL.MinPageCount	Mindestzahl der Druckseiten
LL.ProjectDescription	Projektbeschreibung
LL.SlideShow.TransformationID	Transformation für Präsentationsmodus
LL.MAIL.ShowDialog	Maildialog anzeigen

### Parameter:

**Zeichenkette** Name des Projekt-Parameters  
**Boolean** (optional) Bestimmt ob der Rückgabewert direkt zurückgegeben werden soll (True), oder berechnet werden soll (False). Voreinstellung: False.

**Rückgabewert:**

**Zeichenkette**

**Beispiel:**

ProjectParameter\$("LL.ProjectDescription")	Ergebnis: "Artikelliste"
ProjectParameter\$("LL.Mail.To")	Ergebnis: "name@combit.de"

### **ProjectPath\$**

---

**Aufgabe:**

Gibt den Pfad der Projektdatei an, optional inklusive des Dateinamens (sonst mit "\" am Ende).

**Parameter:**

**Boolean** (optional) Gibt an, dass der Pfad inklusive des Dateinamens zurückgeliefert wird (True). (Voreinstellung False).

**Rückgabewert:**

**Zeichenkette**

**Beispiel:**

ProjectPath\$ ()	Ergebnis: "C:\Programme\LL\"
ProjectPath\$ ()	Ergebnis: "C:\Programme\LL\x.lst"

### **Quarter**

---

**Aufgabe:**

Bestimmt die Quartalszahl eines Datums und gibt diese als Zahl zurück.

**Parameter:**

**Datum**

**Boolean** (optional) Bestimmt, ob die Berechnung Quartal relativ zum Jahr (1..4) oder absolut seit 01.01.0001 (1..) zurückgegeben werden soll. Voreinstellung: False (relativ).

**Rückgabewert:**

**Zahl**

**Beispiel:**

Quarter(Date("04.07.1776"))	Ergebnis: 3
Quarter(Date("04.07.1776"), True)	Ergebnis: 7103

## RainbowColor

---

### Aufgabe:

Berechnet einen Farbwert zwischen Violett und Rot entsprechend des Wertes des ersten Parameters. Z.B. für Regenbogenfarben in Crosstabs.

### Parameter:

**Zahl** Wert, der dargestellt werden soll.  
**Zahl** Grenzwert für Violett.  
**Zahl** Grenzwert für Rot

### Rückgabewert:

**Zahl**

## RegExMatch\$

---

### Aufgabe:

Gibt den einen Teil der Zeichenkette zurück, der der Regular Expression entspricht oder die Gruppe entsprechend dem 3. Parameter.

Die Regular Expression entspricht dabei der Pearl 5 Syntax, sowie in den meisten Details der Regular Expression-Syntax der Visual Basic Scripting Engine.

### Parameter:

**Zeichenkette**  
**Zeichenkette**  
**Number**

### Rückgabewert:

**Zeichenkette**

### Beispiele:

RegExMatch\$ ("1234xyz5678", "[0-9]+")      Ergebnis: "1234"

## RegExSubst\$

---

### Aufgabe:

Ersetzt die Teile des ersten Arguments, bei denen der reguläre Ausdruck zutrifft, durch einen neuen Wert.

### Parameter:

**String** die zu überprüfende Zeichenkette  
**String** regulärer Ausdruck  
**String** Ersetzungsausdruck (kann "\0" für den gesamten Match, oder "\1"... "\9" für die entsprechende Gruppe enthalten)

**Boolean** (optional) Gibt an, ob nur das erste Vorkommen ersetzt werden soll. Voreinstellung: False

**Rückgabewert:**

**Zeichenkette**

**Beispiel:**

RegExSubSt\$("1234xyz6789","[0-9]+", "a") Ergebnis: "axyza"

RegExSubst\$("1234xyz6789","[0-9]+", "a", True) Ergebnis: "axyz6789"

## RemainingTableSpace

---

**Aufgabe:**

Gibt den in dem Tabellenobjekt für Daten und Gruppenzeilen zur Verfügung stehenden Platz zurück. Der Parameter gibt die Einheit des Rückgabewerts an. Diese Funktion kann verwendet werden, um bedingte Seitenumbrüche vor Gruppenzeilen durchführen, z.B. "Wenn nur 5% Platz übrig, dann vorher umbrechen".

**Parameter:**

**Boolean** (optional) True: der Wert ist in SCM-Einheiten (1/1000mm), False: der Wert ist der Prozentwert der gesamten Tabellengröße (Voreinstellung: False)

**Rückgabewert:**

**Zahl**

## Rep\$

---

**Aufgabe:**

Gibt eine Zeichenkette zurück, die aus der entsprechenden Anzahl Zeichen des ersten Parameters gebildet wird.

**Parameter:**

**Zeichenkette**

**Zahl**

**Rückgabewert:**

**Zeichenkette**

**Beispiele:**

Rep\$ ("-",10) Ergebnis: "-----"

Rep\$ ("+-",5) Ergebnis: "+-+-+-+-"

## RGB

---

**Aufgabe:**

Berechnet einen Farbwert anhand relativer Rot-, Grün- und Blau-Anteile (zwischen 0 und 255). Dabei bedeutet 0 kein Farbwert und 255 der maximale Farbwert. Diese Funktion kann verwendet werden, um z.B. die Schriftfarbe in Textobjekten per Formel zu bestimmen.

**Parameter:**

**Zahl**                    Rotanteil  
**Zahl**                    Grünanteil  
**Zahl**                    Blauanteil

**Rückgabewert:**

**Zahl**

**Beispiel:**

Cond(Betrag<0, RGB(255,0,0), RGB(0,0,0))

---

**Right\$**

---

**Aufgabe:**

Kürzt eine Zeichenkette von links um so viele Zeichen, dass gerade noch die unter Zahl angegebene Anzahl von Zeichen übrigbleibt. Wenn die ursprüngliche Zeichenkette schon klein genug ist, wird diese nicht verändert.

**Parameter:**

**Zeichenkette**

**Zahl**

**Boolean** (optional) True: Der abgeschnittene Wert beginnt mit "..."  
(Voreinstellung False). Bei Zahl < 3 wird die Einstellung ignoriert.

**Rückgabewert:**

**Zeichenkette**

**Beispiel:**

Right\$ ("ABCD", 2)                    Ergebnis: "CD"

---

**Round**

---

**Aufgabe:**

Rundet den Wert einer Zahl auf die im zweiten Argument angegebene Anzahl Nachkommastellen. Die voreingestellte Anzahl Nachkommastellen ist 0.

**Parameter:**

**Zahl**

**Zahl** (optional)

**Rückgabewert:**

### Zahl

#### Beispiele:

Round(3.1454,2)	Ergebnis: 3,15
Round(2.5)	Ergebnis: 3

### RTrim\$

---

#### Aufgabe:

Entfernt Leerzeichen am Ende einer Zeichenkette.

#### Parameter:

Zeichenkette

#### Rückgabewert:

Zeichenkette

#### Beispiel:

RTrim\$ ("Hallo Welt ") Ergebnis: "Hallo Welt"

### Second

---

Bestimmt die Sekunde des übergebenen Datums und gibt sie als Zahl zurück. Wenn der Parameter weggelassen wird, wird die Sekunde der Druckzeit zurückgeliefert.

#### Parameter:

Datum (optional)

#### Rückgabewert:

Zahl

### SetVar

---

Setzt einen Wert in den Variablenspeicher für eine spätere Verwendung über die GetVar()-Funktion. Die Aufgabe dieser Funktionen ist die eines einfachen Zwischenspeichers für Werte. Sie sollten keine komplexen Verschachtelungen mit GetVar/SetVar durchführen oder die beiden Funktionen miteinander kombinieren - hier können, insbesondere in Kopf-, Fuß- und Gruppenzeilen, unerwartete Effekte auftreten.

#### Parameter:

Zeichenkette

Alle

#### Rückgabewert:

Alle

#### Beispiel:

SetVar ("Seite", Page())

## Sign

---

### Aufgabe:

Gibt das Vorzeichen des Werts zurück (+1 für positiven Wert, -1 für negativen Wert, oder 0, wenn der Wert 0 ist).

### Parameter:

Zahl                      Wert

### Rückgabewert:

Zahl

### Beispiel:

Sign (-3)                      Ergebnis: -1

## Sin

---

### Aufgabe:

Berechnet den Sinus des Werts (in Grad, wenn keine andere Einheit gewählt ist).

### Parameter:

Zahl                      Wert

Zahl                      (optional) Modus (0=Grad, 1=Radiant). Voreinstellung: 0.

### Rückgabewert:

Zahl

### Beispiel:

Sin (90)                      Ergebnis: 1

## Sqrt

---

### Aufgabe:

Gibt die Quadratwurzel einer Zahl zurück.

### Parameter:

Zahl

### Rückgabewert:

Zahl

### Beispiel:

Sqrt (4)                      Ergebnis: 2

## StartsWith

---

### Aufgabe:

Testet, ob die Zeichenkette im ersten Argument mit der Zeichenkette im zweiten Argument beginnt.

### Parameter:

**Zeichenkette**

**Zeichenkette**

**Boolean** Gibt an, ob Groß-/Kleinschreibung beachtet werden soll.  
Voreinstellung: False

### Rückgabewert:

**Boolean**

### Beispiele:

StartsWith ("Hallo Welt", "hal")

Ergebnis: True

StartsWith ("Hallo Welt", "elt")

Ergebnis: False

## StdDeviation

---

### Aufgabe:

Liefert die Standardabweichung der Datenmenge, die sich aus dem ersten Argument ergibt.

### Parameter:

**Zahl**

**Boolean** (optional) True: nach der Ausgabe werden die für die Berechnung gemerkten Werte gelöscht (Voreinstellung: True). Bitte beachten Sie, dass die für die Berechnung gemerkten Werte generell bei jedem (Unter-)Tabellenende gelöscht werden. Der zweite Parameter bestimmt lediglich, ob die Werte bei einer Ausgabe schon innerhalb der Tabelle gelöscht werden

### Rückgabewert:

**Zahl**

### Beispiel:

StdDeviation (Anzahl \* Stückpreis)

## Str\$

---

### Aufgabe:

Wandelt eine Zahl in eine Zeichenkette. Die Zahl wird mit 6 Nachkommastellen formatiert, dabei eventuell gerundet. Die Länge ist variabel.

**Parameter:****Zahl****Zahl**

(optional) Gibt die Länge der gewünschten Zeichenkette an. Wenn die Zahl für dieses Format jedoch zu groß ist, kann die resultierende Zeichenkette länger als gewünscht werden. Wenn die Zahl zu klein ist, werden Leerstellen angehängt, je nach Vorzeichen rechts (negativ) oder links (positiv).

**Zahl**

(optional) Gibt die Zahl der Nachkommastellen an. Ist er positiv, wird die Zahl als Gleitkommazahl dargestellt, ist er negativ, in wissenschaftlicher Schreibweise. Wenn kein dritter Parameter angegeben ist, wird die Zahl der Nachkommastellen bei einem Ganzzahlwert auf 0 und bei einem Fließkommawert aus Kompatibilitätsgründen auf 5 gesetzt.

**Rückgabewert:****Zeichenkette****Beispiele:**

Str\$(Pi)

Ergebnis: "3.141593"

Str\$(Pi,3)

Ergebnis: "3"

Str\$ (-Pi,12,-3)

Ergebnis: "-3.141e+00"

**StrPos****Aufgabe:**

Liefert die Position des n-ten Vorkommens einer Suchzeichenkette in einer Zeichenkette zurück. Über einen dritten Parameter kann mitgegeben werden, das wievielte Vorkommen des Suchbegriffs in der Zeichenkette zurückgegeben werden soll. Der Default ist 1.

Das erste Zeichen der Zeichenkette entspricht Position 0.

-1 als Rückgabewert bedeutet, dass die Zeichenkette nicht (mehr) vorkommt.

**Parameter:****Zeichenkette****Zeichenkette**

Such-Zeichenkette

**Zahl**

(optional)

**Rückgabewert:****Zahl****Beispiel:**

StrPos ("Hallo Welt","o")

Ergebnis: 4

## StrRPos

---

### Aufgabe:

Liefert die Position einer Suchzeichenkette in einer Zeichenkette zurück. Es wird von hinten her gesucht. Über den dritten Parameter kann mitgegeben werden, das wievielte Vorkommen des Suchbegriffs von hinten gesucht werden soll. Die Voreinstellung ist 1.

Das erste Zeichen der Zeichenkette entspricht Position 0.

-1 als Rückgabewert bedeutet, dass die Zeichenkette nicht (mehr) vorkommt.

### Parameter:

**Zeichenkette**

**Zeichenkette**    Such-Zeichenkette

**Zahl**            (optional)

### Rückgabewert:

Zahl

### Beispiele:

StrRPos ("Hallo Welt","a")

Ergebnis: 1

## StrSubst\$

---

### Aufgabe:

Durchsucht eine Zeichenkette nach Vorkommen einer Such-Zeichenkette und ersetzt ihn durch die Zeichenkette des dritten Arguments (Ersetzungs-Zeichenkette). Ist kein drittes Argument vorhanden, wird die dem zweiten Argument entsprechende Zeichenkette einfach gelöscht.

### Parameter:

**Zeichenkette**

**Zeichenkette**    Such-Zeichenkette

**Zeichenkette**    (optional) Ersetzungs-Zeichenkette

### Rückgabewert:

Zeichenkette

### Beispiel:

StrSubst\$ ("Hallo Welt","ll","nn")

Ergebnis: "Hanno Welt"

## Sum

---

### Aufgabe:

Liefert die Summe des ersten Argumentes. Aggregatsfunktionen arbeiten grundsätzlich tabellenspezifisch.

**Hinweis:** Summenvariablen (siehe Kapitel "Arbeiten mit Summenvariablen") sind eine alternative Möglichkeit Summen und Zähler zu bilden. Summenvariablen arbeiten grundsätzlich tabellenübergreifend.

**Parameter:****Zahl****Boolean**

(optional) True: nach der Ausgabe werden die für die Berechnung gemerkten Werte gelöscht (Voreinstellung: True). Bitte beachten Sie, dass die für die Berechnung gemerkten Werte generell bei jedem (Unter-)Tabellenende gelöscht werden. Der zweite Parameter bestimmt lediglich, ob die Werte bei einer Ausgabe schon innerhalb der Tabelle gelöscht werden.

**Rückgabewert:****Zahl****Beispiel:**

Sum (Stückpreis)

Ergebnis: Summe Stückpreise

**Tan****Aufgabe:**

Berechnet den Tangens des Wertes (in Grad, wenn keine andere Einheit gewählt ist).

**Parameter:****Zahl**

Wert

**Zahl**

(optional) Modus (0=Grad, 1=Radient). Voreinstellung: 0.

**Rückgabewert:****Zahl****Beispiel:**

Tan (45)

Ergebnis: 1,00

**Time\$****Aufgabe:**

Liefert die aktuelle Zeit, die über eine Zeichenkette formatiert wird.

Folgende Formatschalter stehen zur Verfügung:

Schalter	Bedeutung
%h	Stunden im 12-Stundenformat
%H	Stunden im 24-Stundenformat

## Übersicht der Funktionen

---

%m oder %i	Minuten
%s	Sekunden
%P	Anzeige der Tageshälfte (A.M. / P.M.)
%p	Anzeige der Tageshälfte (a.m. / p.m.)

**Parameter:**

**Zeichenkette**

**Rückgabewert:**

**Zeichenkette**

**Beispiel:**

Time\$ ("%02h:%02m:%02s")

## Today

---

**Aufgabe:**

Gibt das aktuelle Datum im Format "tt.mm.jjjj" zurück.

**Parameter:** -

**Rückgabewert:**

**Datum**

**Beispiel:**

Today()

## Token\$

---

**Aufgabe:**

Gibt die n-te Zeichenkette aus dem ersten Argument zurück, das durch den Separator des dritten Parameters in einzelne Zeichenketten getrennt ist.

**Parameter:**

**Zeichenkette**

**Zahl** <n>, d.h. Position der Zeichenkette (0-basiert)

**Zeichenkette** Trennzeichen

**Zeichenkette** (optional) Zeichenkette die zurückgegeben wird, falls Wert außerhalb

**Rückgabewert:**

**Zeichenkette**

**Beispiel:**

Token\$ ("ABC;DEF;GHI;JKL",2,";")      Ergebnis: "GHI"

## ToRTF\$

---

### Aufgabe:

Liefert eine Zeichenkette im RTF-Format zurück. Das ist deshalb notwendig, weil Zeichenketten möglicherweise eines der im RTF-Format speziell definierten Symbole ('\\', '{' oder '}') enthalten können. Aus Kompatibilitätsgründen arbeitet die Funktion erst dann wie beschrieben, wenn der optionale zweite Parameter True ist.

### Parameter:

#### Zeichenkette

**Boolean** (optional) erzwingt das Ersetzen von RTF-Tokens zu Plaintext (Voreinstellung: False)

### Rückgabewert:

#### Zeichenkette

### Beispiel:

Wenn beispielsweise das Feld PRODCODE eins der Zeichen enthalten könnte, dann müsste man den Text folgendermaßen einfügen:

"<<ANREDE>> <<NAME>>, Sie haben unser Produkt <<PRODUKT>>, Code <<ToRTF\$(PRODCODE)>> erhalten..."

## Total

---

### Aufgabe:

Berechnet den Ausdruck im Argument für das gesamte Kreuztabellen-Objekt (z.B. kategorienübergreifend die Summe aller Umsätze).

### Parameter:

#### Alle

### Rückgabewert:

#### Alle

### Beispiel:

Prozentualer Umsatzanteil einer Kategorie:

$\text{Sum}(\text{Umsatz})/\text{Total}(\text{Sum}(\text{Umsatz}))*100$

## TotalPages\$

---

### Aufgabe:

Liefert die Gesamtzahl der Seiten. Die zurückgegebene Zeichenkette wird beim Ausdruck durch die Gesamtzahl der Seiten ersetzt. Diese Funktion dient nur der Anzeige und kann nicht in Formeln verwendet werden. Eine Berechnung der Formel `val(TotalPages$)` ist nicht möglich.

Bitte beachten Sie, dass sich bei Verwendung dieser Funktion das Zeitverhalten beim Ausdruck ändern kann. Ein eventuell vorhandener Fortschrittsbalken wird schneller die 100% erreichen, durch die notwendige Nachbearbeitung der Ausgabe kann sich dann allerdings eine Verzögerung bis zur tatsächlichen Anzeige des Druckergebnisses ergeben. Beachten Sie, dass bei vieltausendseitigen Projekten eine Verwendung dieser Funktion nicht sinnvoll ist, da die Seiten der Ausgabe zwischengespeichert werden müssen, was ggf. die verfügbaren Hardwaremöglichkeiten nicht erlauben.

### Parameter

**Zahl** (optional) einzuberechnendes Offset, d.h. dieser Wert wird zur Gesamtzahl der Seiten addiert. Voreinstellung: 0.

### Rückgabewert:

**Zeichenkette**

### Beispiel:

"Seite "+Page\$ ()+"/"+"TotalPages\$ () Ergebnis: "Seite 1/3"

"Seite "+Page\$ ()+"/"+"TotalPages\$ (1) Ergebnis: "Seite 1/4"

## Translate\$

---

### Aufgabe:

Übersetzt den übergebenen Text, sofern er in dem von der Applikation übergebenen Wörterbuch enthalten ist.

### Parameter:

**Zeichenkette**

### Rückgabewert:

**Zeichenkette**

### Beispiel:

Translate\$("Page {0} of {1}", Page\$(), TotalPages\$())

Ergebnis in deutscher Sprache z.B.: Seite 1 von 2

## UnitFromSCM

---

### Aufgabe:

Wandelt eine SCM-Einheit (1/1000 mm) in die Druckeinheit (inch/mm) um. Wichtig, um Eigenschaftswerte unabhängig von der gewählten Druckeinheit zu machen.

### Parameter:

**Zahl**

### Rückgabewert:

**Zahl**

**Beispiel:**

```
Cond(Page)=1,UnitFromSCM(100000),UnitFromSCM(20000))
```

**Upper\$**

---

**Aufgabe:**

Wandelt die Zeichen einer Zeichenkette in Großbuchstaben um.

**Parameter:**

**Zeichenkette**

**Rückgabewert:**

**Zeichenkette**

**Beispiel:**

```
Upper$ ("hallo welt")
```

Ergebnis: "HALLO WELT"

**Val**

---

**Aufgabe:**

Die Zeichenkette wird als Zahl interpretiert und in eine solche gewandelt. Wenn ein Fehler auftritt, ist das Resultat 0. Das Dezimalzeichen muss immer als "." angegeben werden.

**Parameter:**

**Zeichenkette**

**Rückgabewert:**

**Zahl**

**Beispiele:**

```
Val ("3.141")
```

Ergebnis: 3.141

```
Val ("3,141")
```

Ergebnis: 3

```
Val ("3.141e2")
```

Ergebnis: 314.2

```
Val (ChrSubst$ ("3,141", ",", "."))
```

Ergebnis: 3.141

**Variance**

---

**Aufgabe:**

Liefert die Varianz der Datenmenge, die sich aus dem ersten Argument ergibt.

**Parameter:**

**Zahl**

**Boolean**

(optional) True: nach der Ausgabe werden die für die Berechnung gemerkten Werte gelöscht (Voreinstellung: True). Bitte beachten Sie, dass die für die Berechnung

gemerkten Werte bei jedem (Unter-)Tabellenende gelöscht werden. Der zweite Parameter bestimmt lediglich, ob die Werte bei einer Ausgabe schon innerhalb der Tabelle gelöscht werden.

**Rückgabewert:**

Zahl

**Beispiel:**

Variance (Anzahl \* Stückpreis)

## Woy

---

**Aufgabe:**

Gibt die Wochennummer des übergebenen Datums zurück.

**Parameter:**

Datum

Zahl

(optional) Der optionale zweite Parameter bestimmt, wie die erste Woche des Jahres bestimmt wird.

---

0	Woche mit dem ersten Arbeitstag (Voreinstellung)
---	--

---

1	Woche des 1. Januars
---	----------------------

---

2	Erste Woche mit mindestens 4 Tagen
---	------------------------------------

---

3	Erste Woche mit 7 Tagen
---	-------------------------

---

4	Woche mit dem ersten Montag
---	-----------------------------

---

**Rückgabewert:**

Zahl

**Beispiel:**

Woy (Date("04.07.1776"),0)

Ergebnis: 27

## Year

---

**Aufgabe:**

Bestimmt das Jahr eines Datums und gibt es als Zahl zurück.

**Parameter:**

Datum

**Rückgabewert:**

Zahl

**Beispiel:**

Year (Date("04.07.1776"))

Ergebnis: 1776

## **Year\$**

---

### **Aufgabe:**

Bestimmt das Jahr eines Datums und gibt es als Zeichenkette zurück.

### **Parameter:**

Datum

### **Rückgabewert:**

Zeichenkette

### **Beispiel:**

Year\$ (Date("04.07.1776"))

Ergebnis: "1776"