

# ADAM

## Tabellenkalkulation für Windows

Diese Einführung in **Adam** ist absichtlich so knapp wie möglich gehalten. Wir setzen voraus, daß Sie mit den Grundzügen der Arbeit mit Tabellenkalkulationen vertraut sind. Diese Einführung soll es Ihnen ermöglichen, so schnell wie möglich mit den Eigenheiten von **Adam** bekannt zu machen damit Sie ohne großen Aufwand in die Praxis kommen.

Falls Sie noch nie mit einer Tabellenkalkulation gearbeitet haben brauchen Sie vielleicht etwas Hilfe durch einen Freund oder durch eines der vielen einschlägigen Fachbücher über dieses Thema.

### Teil1: Grundlegendes über Adam

HIER BILD STATUSZEILE EINFUEGEN!!!

Das Fenster von Adam wird im wesentlichen vom *Arbeitsblatt* eingenommen. Das Arbeitsblatt besteht aus einer quadratischen Anordnungen von *Zellen*. Die Zellen sind die elementaren Bestandteile des von Adam. Adam hat 255 Zeilen mit je 255 Zellen, oder - anders ausgedrückt - 255 Reihen und 255 Spalten. Die Reihen sind durchnummeriert, die Spalten werden zunächst mit a - z bezeichnet, dann mit zwei Buchstaben aa - iu. Durch die Kombination des alphabetischen Spaltennamens a - iu mit dem numerischen Reihennamen 1-255 wird eine Zelle im Arbeitsblatt eindeutig identifiziert, z. B. a1 - die Zelle in der Ecke links oben - oder b3 oder ab34 usw..

Von den 255 x 255 Zellen des Arbeitsblatts ist immer nur ein Ausschnitt auf dem Bildschirm zu sehen. Dieser Ausschnitt ist mit den Laufleisten (oder Rollbalken) am rechten bzw. unteren Fensterrand zu verschieben.

Von den Zellen ist jeweils eine als die aktive Zelle herausgehoben. Sie hat an Ihrem inneren Rand zusätzlich ein fein punktiertes Rechteck und ist zudem schwarz hinterlegt. Wie Sie später sehen werden ist aber das punktierte Rechteck das ausschlaggebende Kennzeichen. Außerdem wird die Bezeichnung der aktiven Zelle in der *Statuszeile* direkt unter dem Menu angezeigt.

In der *Statuszeile* wird auch der interne Zelleninhalt angezeigt falls die Zelle nicht leer ist. Dieser ist der eigentliche Inhalt der Zelle. Auf dem Arbeitsblatt wird der externe Zelleninhalt angezeigt. Der externe Inhalt unterscheidet sich vom internen in verschiedener Weise:

- ☐ Die Zelle kann eine Formel beinhalten. Dann wird in der Statuszeile die Formel angezeigt während auf dem Arbeitsblatt das Ergebnis angezeigt wird.

- ☐ Die Darstellung auf dem Arbeitsblatt ist formatiert. Sie muß sich an der Größe der aktiven Zelle auf dem Arbeitsblatt orientieren. Zahlen werden mit einer bestimmten Zahl von Nachkommastellen ausgegeben.

Zur Darstellung von *Programmen* finden Sie mehr unter dem entsprechenden Punkt der Dokumentation.

Beim Start zeigt Ihnen Adam entweder ein leeres Arbeitsblatt oder das zuletzt bearbeitete

Arbeitsblatt an. Im Titel des Fensters erscheint der Programmname 'Adam' zusammen mit dem Namen der geladenen Arbeitsblattdatei oder der Bezeichnung NEU in rechteckigen Klammern. Der sichtbare Ausschnitt ist der linke obere Teil des Arbeitsblatts. Die Zelle in der linken oberen Ecke ist die aktive Zelle.

### **Bewegen der aktiven Zelle**

Mit den Pfeiltasten kann die aktive Zelle jeweils um eine Position innerhalb der Reihen oder Spalten verschoben werden. Dabei werden in der *Statuszeile* Zellbezeichnung und interner Zellinhalt entsprechend verändert. Wird die aktive Zelle über den sichtbaren Ausschnitt hinwegbewegt so wird automatisch der Ausschnitt verschoben.

Wird die linke Maustaste gedrückt so wird die Zelle unter dem Mauszeiger zur aktiven Zelle.

Beim Verschieben des Ausschnitts mit den Laufleisten wird die aktive Zelle nicht bewegt.

### **Selektion von Zellen**

Einige Menüfunktionen von Adam arbeiten nicht oder nicht nur mit einzelnen Zellen sondern mit einer Gruppe von Zellen, z.B. die Funktionen des Menus `\Bearbeiten`. Auf welchen Zellen diese arbeiten wird durch die *Selektion* festgelegt. Man selektiert Zellen indem man den Mauszeiger bei gedrückter linker Maustaste über die gewünschten Zellen bewegt. Dabei wird zunächst ein *Gummiband* um die Zellen gelegt; nach Loslassen der linken Maustaste verschwindet das Gummiband und die selektierten Zellen werden schwarz hinterlegt.

Eine dieser Zellen ist nach wie vor die aktive Zelle, nämlich die mit dem punktierten Rechteck.

Die aktive Zelle ist immer selektiert, daher ist sie immer schwarz hinterlegt.

Diese Selektion kann auch durch Benutzen der Pfeiltasten bei niedergehaltener `SHIFT`-Taste gemacht werden.

Man kann eine Spalte oder Reihe als Ganzes selektieren, indem man mit der linken Maustaste auf den Spalten- bzw. Reihennamen klickt. Mehrere Spalten oder Reihen selektiert man durch Bewegen des Mauszeigers bei gedrückter linker Maustaste über mehrere Spalten- oder Reihennamen hinweg. Ein äquivalente Tastaturbedienung gibt es hierbei nicht.

### **Bearbeiten des Zellinhalts**

Die Eingabe oder Veränderung des Zellinhalts erfolgt über das Feld mit dem internen Zellinhalt in der *Statuszeile*, dieses Feld ist somit gleichzeitig das *Eingabefeld* für die aktive Zelle. Die Eingabe kann auf zwei Arten gestartet werden, entweder durch Drücken der linken Maustaste während sich der Mauszeiger über dem Eingabefeld befindet oder durch Drücken einer Taste auf der Tastatur (Pfeiltasten und andere Steuerungselemente natürlich ausgenommen).

Mit der linken Maustaste setzt man eine Schreibmarke in das Eingabefeld. Bei bereits vorhandenem Inhalt kann man diese gezielt an eine Stelle setzen und von dort aus Änderungen vornehmen. Der zuvor invers dargestellte Zellinhalt wird nun normal angezeigt, Eingaben ergänzen das Vorhandene.

Wird die Eingabe durch Anschlag auf die Tastatur begonnen, wird der vorhandene Zellinhalt überschrieben, es sei denn die Dateneingabe wird mit den Steuerungstasten `Pos1` oder `Ende` begonnen. In diesem Fall gilt dasselbe wie bei Start mit der Maus. Andernfalls wird der Zellinhalt

für eine Neueingabe gelöscht.

Solange die Eingabe andauert bewegen die Pfeiltasten die Schreibmarke innerhalb des Zellinhalts und nicht die aktive Zelle über das Arbeitsblatt.

Im übrigen verhält sich das Eingabefeld Windowsüblich. Beispielsweise lassen sich mit **SHIFT**+Pfeil oder gedrückter linker Maustaste Teile selektieren und löschen usw..

Die Dateneingabe wird durch die Tasten **Enter** (CR) oder **ESC** beendet. Bei **Enter** wird der Zellinhalt endgültig übernommen und in seiner externen Darstellung auf dem Arbeitsblatt angezeigt. Bei **ESC** wird die Dateneingabe verworfen und der ursprüngliche Zellinhalt wiederhergestellt. In beiden Fällen bewegen die Pfeiltasten jetzt wieder die aktive Zelle über das Arbeitsblatt.

## Zelltypen

Jede Zelle kann einen Inhalt folgender Art aufnehmen:

- ☐ **Kommazahl:** Werden Ziffern eingegeben, eventuell zusammen mit einem Dezimalkomma und Vorzeichen + und -, wird die Eingabe als Zahl interpretiert und auf dem Arbeitsblatt rechtsbündig dargestellt. Die Eingabe kann auch in wissenschaftlicher Form mit den Exponentenkennzeichen E und D erfolgen, sie wird jedoch in die normale Form umgewandelt.
- ☐ **Text:** Tippt man alphanumerische Zeichen ein, wird die Eingabe als Zeichenkette interpretiert und linksbündig im Arbeitsblatt dargestellt. Ist das erste Zeichen eine Ziffer, so muß vor der eigentlichen Zeichenkette ein Anführungszeichen " eingegeben werden, damit die Eingabe nicht als Zahl interpretiert wird. Das Anführungszeichen selbst wird nicht mit übernommen.
- ☐ **Datum:** Werden Ziffern und Punkte in der Form tt.mm.jjjj oder tt.mm.jj eingegeben wird die Eingabe als Datum interpretiert. Die Jahreszahl muß zwischen 1900 und 2078 liegen. Zweistellige Jahresangaben zwischen 0 und 99 werden auf 19xx erweitert. Angaben für Monat und Tag werden geprüft. Bei Fehleingaben wie z.B. 32.1.93 wird eine Warnung ausgegeben.
- ☐ **Zeit:** Werden Ziffern und ein Doppelpunkt in der Form hh:mm eingegeben wird die Eingabe als Zeit interpretiert. Sowohl die Stunde als auch die Minute werden auf Plausibilität geprüft, gegebenenfalls wird eine Warnung ausgegeben, z.B. bei 25:30.
- ☐ **Formel:** wird als erstes Zeichen ein Gleichheitszeichen = eingegeben, so wird der Zellinhalt als Formel angesehen. Die Formel wird berechnet - falls sie fehlerfrei eingegeben wurde - und das Ergebnis auf dem Arbeitsblatt dargestellt.
- ☐ **Programm:** wird ein Zeichen @ eingegeben, so handelt es sich um eine Programmzelle. Da zur Eingabe von Programmen eine Zeile nicht ausreicht ist für die eigentliche Programmeingabe eine andere Vorgehensweise nötig. Durch jede weitere Eingabe von Enter oder durch einen Doppelklick der linken Maustaste wird für die aktive Zelle ein Editorfenster geöffnet, in welche das Programm bearbeitet werden kann. Weitere Einzelheiten entnehmen Sie bitte dem Abschnitt über Programmierung. Der vordergründige Zellinhalt @ kann aus Sicherheitsgründen nicht ohne weiteres überschrieben werden, er muß durch die Menüfunktion /Bearbeiten/Radieren entfernt werden.

Der maximale Zellinhalt ist 255 Zeichen. Da nicht alle gleichzeitig in der Statuszeile sichtbar sein

können wird ein Ausschnitt durchgerollt.

Zahlen werden auf dem Arbeitsblatt standardmäßig mit 2 Nachkommastellen und rechsbündig dargestellt. Text, Datum und Zeit werden standardmäßig linksbündig dargestellt, lassen sich aber umformatieren (`\Bearbeiten\Spalte\Format`). Zellen die Formeln oder Programme enthalten werden nach dem Type des Ergebnisses formatiert.

Zellen mit Datum oder Zeit werden intern als Kommazahl gehalten und nur extern speziell formatiert.

## Formeln

Wird als erstes Zeichen ein Gleichheitszeichen = eingegeben wird der Zellinhalt als Formel interpretiert. Eine Formel berechnet aus mehreren Argumenten unter Anwendung geeigneter Operatoren ein Ergebnis. Formeln können entweder numerische Formeln oder Textformeln sein. Eine Kombination zwischen numerischen Argumenten und Zeichenketten ist immer ein Fehler.

Numerische Formeln berechnen ein numerisches Ergebnis aus numerischen Argumenten und numerischen Rechenfunktionen unter Anwendung der mathematischen Operatoren +, -, \* und /. Die Argumente sind Zahlenkonstanten, numerische Zellinhalte und freie Variablen. Letztere haben nur bei Programmen Bedeutung. Beispiele:

[a1] 12,1  
[a2] =a1/2  
[a3] =wurzel(2,0)-1,0  
[a4] =a2+a3

Wird die Formel =a1/2 in die Zelle a2 eingegeben (angedeutet durch die Angabe in rechteckigen Klammern), so wird Zelle a2 der halbe Wert von Zelle als Ergebnis zugewiesen, also hier 6,05. Dieses Ergebnis wird auf dem Arbeitsblatt angezeigt. Entsprechend erhält Zelle a3 die quadratische Wurzel der Zahl 2 vermindert um 1 als Ergebnis zugewiesen, *wurzel* ist eine numerische Rechenfunktion.

Textformeln bilden als Ergebnis eine Zeichenkette aus anderen aus. Die Argumente können konstante Zeichenketten, textliche Inhalte von anderen Zellen oder Textfunktionen sein. Als einziger Operator ist die *Verkettung* + zugelassen. Die *Verkettung* ist nicht zu verwechseln mit dem numerischen Operator +, sie benutzt lediglich das gleiche Symbol. Die *Verkettung* addiert nicht zwei Werte sondern bildet eine neue Zeichenkette durch Aneinanderreihen von Teilketten. In einer Formel werden konstante Zeichenketten immer in Anführungszeichen gesetzt, auch wenn sie nur alphanumerische Zeichen enthält. Beispiele:

[a5] Test  
[a6] ="Ein "+a5'  
[a7] =links(a6';3)

Zelle a6 erhält als Ergebnis die Zeichenkette "Ein Test" zugewiesen. Die Teilkette "Test" wird aus der Zelle a5 ausgelesen, diese enthält Text. Als Kennzeichen, daß die Zelle als Zeichenkette verarbeitet werden soll, ist die Zellbezeichnung mit angehängten Apostroph als a5' geschrieben. Andernfalls würde a5 als numerisches Argument interpretiert, was in Verbindung mit dem Textargument "Ein " als Fehler gemeldet würde. Zelle a7 erhält als Ergebnis die 3 ersten Zeichen der Zelle a6 zugewiesen, also "Ein". *links* ist eine Textfunktion, sie bildet eine Teilkette aus ihrem ersten Argument. Das erste Argument muß ebenfalls eine Zeichenkette sein, daher ist die Zellbezeichnung a6' wieder mit dem Apostroph versehen um sie als Textargument zu kennzeichnen.

Bitte merken: Soll eine Zelle in einer Formel als numerisches Argument verarbeitet werden, gibt man einfach Ihren Namen an. Soll ihr Inhalt als Zeichenkette akzeptiert und verarbeitet werden, hängt man als Kennzeichen ein Apostroph an. Natürlich muß die Zelle in beiden Fällen auch einen Inhalt vom entsprechenden Typ haben.

Zellen welche ein Datum oder eine Zeit enthalten gelten in Formeln als numerisch, da sie intern als Kommazahlen gespeichert sind. Insbesondere werden Differenzen von Datums- oder Zeitzellen in der Form

=a1-a2

als normale Kommazahlen behandelt und dargestellt.

### Zahlenkonstanten

Zellen enthalten bei numerischem Typ reelle Kommazahlen. Gibt man eine Ganzzahl ohne Komma und Nachkommastellen ein wird diese automatisch in eine Kommazahl gewandelt. Auch ganze Zahlen als Operanden in Formeln werden automatisch in Kommazahlen gewandelt, allerdings mit einem gewissen Aufwand an Rechenzeit.

Sinn machen ganze Zahlen als Parameter in bestimmten Rechenfunktionen. Die oben bereits erwähnte Rechenfunktion *links* zum Beispiel erwartet als zweiten Parameter die Anzahl der zu liefernden Zeichen. Dies ist natürlicherweise eine ganze Zahl. Wird hier eine Kommazahl eingesetzt findet eine automatische Konvertierung in eine Ganzzahl statt; die Nachkommastellen gehen hierbei verloren.

Außerdem werden ganze Zahlen in bestimmten Programmier-elementen verwendet.

### Zelladressierung

In bisher gezeigten Beispielen für Formeln wurde Zellen in den Berechnungen durch ihren Namen *adressiert*, wie z. B. in

[a2] =a1/2

Durch Menüfunktionen im Menü `\Bearbeiten` läßt sich dieser Inhalt der Zelle a2 in andere Zellen kopieren, z.B. um mehrfache Eingabe für gleichartige Berechnungen zu sparen. Durch Selektieren der Zellen a2 bis d2 und die Menüfunktion `\Bearbeiten\Vervielfachen` wird der Inhalt von a2 in die Zelle b2, c2 und d2 kopiert (siehe auch Beschreibung der Menüfunktionen). Danach haben die Zellen a2 bis d2 folgenden Inhalt:

[a2] =a1/2  
[b2] =b1/2  
[c2] =c1/2  
[d2] =d1/2

Beim Vervielfachen wurden also die Zelladressen automatisch geändert. Jede Formel bezieht sich wie in der Ausgangszelle auf das darüberliegende Feld. Das macht Sinn, denn eine viermalige Berechnung desselben Ausdrucks ist im allgemeinen unerwünscht, dagegen eine ähnliche Berechnung wie in der Ausgangszelle wahrscheinlich das, was man haben will. Diese Adressierung von Zellen die beim Kopieren von Zellen verändert wird nennt man *relative Adressierung*.

Manchmal ist es aber gewünscht, daß zumindest Teile der Formel beim Kopieren nicht verändert

werden sollen. Stellen Sie sich bitte vor, Sie hätten in Zelle a1 den aktuellen Mehrwertsteuersatz eingetragen und möchten alle Preise innerhalb Ihres Arbeitsblatts damit beaufschlagen. Dann sollte sich die Adresse 'a1' beim Kopieren von Formeln nicht verändern. Um dieses zu erreichen greifen Sie zur *absoluten Adressierung*. Dazu Schreiben Sie zunächst in Ihr Arbeitsblatt:

[a1] 0,15  
[a2] 1000,0  
[a3] =a2+a2\*\$a\$1

Wenn Sie jetzt die Zellen a3 bis d3 selektieren und a3 darin vervielfachen wird die Adresse von a1 nicht geändert. Tippen Sie nun in die Zellen b2 bis d2 Zahlen ein, werden wird immer mit 15% beaufschlagt und nicht mit Werten aus den (leeren) Feldern b1 bis d1.

Das \$-Zeichen das diese Änderung im Verhalten bewirkt wurde bei a1 sowohl vor die Spaltennamen a als auch vor die Reihennamen 1 gesetzt. Beide Komponenten sollten beim Kopieren unverändert bleiben. Manchmal macht es Sinn beim Kopieren nur eine Komponente der Zelladresse konstant zu halten und die andere veränderlich. Dann wird die Zelladresse beim Kopieren in Spaltenrichtung geändert und beim Kopieren in Reihenrichtung nicht oder genau umgekehrt, jenachdem welche Komponente der Adresse absolut gesetzt wurde. Beispiele für solche *gemischte Adressierung* sind Adressen wie \$b11 oder b\$11.

### Arithmetikfehler

Beim Berechnen von Formeln können Fehler auftreten, beispielsweise Division durch Null. In solchen Fällen wird auf dem Arbeitsblatt kein Ergebnis angezeigt sondern ein Kürzel für die aufgetretene Fehlerart:

- ☐ DIV0: Division durch Null
- ☐ FPERR: Allgemeiner Fehler beim Rechnen mit Kommazahlen
- ☐ ARGUMENT: Fehlerhaftes Argument für Funktion (wurzel(-1) z.B.)
- ☐ PREZIS: Berechnung unpräzise, Kommazahl zu groß oder zu klein

Bitte überprüfen Sie Ihre Formel.

### Automatische Berechnung

Werden Zellen als Operanden in Formeln verwendet, so merkt sich Adam diese Verwendung. Wird eine Zelle durch eine Neueingabe oder auf andere Weise (Zelle enthält selber eine Formel die aufgrund der Änderung einer dritten Zelle neu berechnet wird) geändert, wird jede Zelle welche eine Formel oder ein Programm mit der geänderten Zelle enthält automatisch neu berechnet.

Die Menüfunktion `\Rechnen` ist unter normalen Umständen daher gewöhnlich nutzlos, da alle Zellen stets auf dem neuesten Stand sind. Nur unter bestimmten Umständen beim Programmieren von *benutzerdefinierten Rechenfunktionen* ist sie angebracht (siehe dort).

Das Gedächtnis für die Querverbindungen wurde so bemessen das jede Zelle in circa 50 anderen verwendet werden kann. Nur wenn diese Zahl überschritten wird bewirkt die Anwendung von `\Rechnen` eine Neuberechnung.

### Automatische Sicherung

**Adam** sichert automatisch den Inhalt des Arbeitsblatts in eine Datei 'autosave.adm'. Diese

Sicherung findet nach einer festgelegten Zahl von Tastenanschlägen statt. Die Anzahl der Anschläge und ein Abschalten der Sicherung kann mit der Menufunktion `\Einstellung\Autosicherung` eingestellt werden.

Die Datei 'autosave.adm' ist eine normale Arbeitsblattdatei und kann nach einem Absturz des Systems geladen und unter dem gewünschten Namen gespeichert werden.

## Teil 2: Die eingebauten Rechenfunktionen

Die im Folgenden aufgeführten Rechenfunktionen sind die im Lieferumfang von **Adam** enthaltenen *eingebauten Rechenfunktionen*. Sie sind zu unterscheiden von *benutzerdefinierten Rechenfunktionen*, die im entsprechenden Teil dieser Dokumentation abgehandelt werden.

Die Rechenfunktionen sind entweder vom Typ Numerisch oder von Typ Text und können in Ausdrücken entsprechenden Typs verwendet werden.

Einige von Ihnen schlagen Brücken zwischen beiden Typen von Ausdrücken; sie akzeptieren Argumente von einem Typ und liefern in den anderen Type gewandelten Ergebnisse. Zur Zeit gehört in diese Kategorie allerdings nur die Funktion *Text*.

Wie bereits im Abschnitt 'Formeln' angesprochen gelten auch in Rechenfunktionen die Zelltypen *Datum* und *Text* als numerisch. Einige numerische Funktionen liefern Werte im Datums- oder Zeitformat.

Zeichenketten als Argumente müssen wie in Formeln stets in Anführungszeichen gesetzt werden.

### Bereiche

Als Argumente für Rechenfunktionen kommen außer Komma- und Ganzzahlen und Zelladressen auch *Bereiche* in Frage. Bereiche bezeichnen eine Ansammlung von Zellen, die gleichzeitig durch die Funktion bearbeitet werden sollen. Ein einfacher Bereich besteht aus einem Rechteck benachbarter Zellen. Zwei Eckpunkte des Rechtecks werden zu dessen Definition benutzt, indem die Zelladressen dieses Rechtecks durch einen Doppelpunkt zu dem Bereichsnamen verbunden werden. Beispiel:

a1:c4

Zu dem Bereich a1:c4 gehören die zellen a1, a2, a3, a4, b1, b2, b3, b4, c1, c2, c4. Bereiche sind also für Rechenfunktionen das, was Selektionen (siehe Abschnitt 'Selektion') für Menufunktionen sind.

Eine einfache Zelladresse wie z.B. a1 ist der Sonderfall eines 1-Zell-Bereichs. Wo eine Rechenfunktion einen Bereich als Argument erwartet sind daher natürlich auch einzelne Zellen zugelassen.

Einfache Bereiche können zu *zusammengesetzten* Bereichen kombiniert werden. Dazu verbindet man einfache Bereichsnamen mit '&' zu einem Verbund. Beispiele:

a1:c4&d2:d4  
d2:d4&c4  
a1&a3

Wie das zweite und insbesondere das dritte Beispiel zeigen, können Bereiche auch eine Ansammlung einzelner Zellen sein.

Außer diesen direkten Zellbezeichnungen kann man auch mit Bereichsnamen arbeiten. Die Menufunktion `\Bearbeiten\Bereichsname` erlaubt die Zuordnung von Namen zu Zellbereichen.

## Übersicht der Rechenfunktionen nach Funktionskreisen:

### Datums- und Zeitfunktionen

heute	Aktuelles Datum
datum	Datumswert einer Zahl XXXX
monat	Monat eines Datumswerts
jahr	Jahr eines Datumswerts
tag	Tag eines Datumswerts
jetzt	Aktuelle Zeit
zeit	Zeitwert einer Zahl
stunde	Stunde eines Zeitwerts
minute	Minute eines Zeitwerts

### Statistische Funktionen

summe	Summe eines Bereichs
min	Minimum eines Bereichs
max	Maximum eines Bereichs
mittel	Mittelwert eines Bereichs
varianz	Varianz eines Bereichs

### Mathematische Funktionen

pi	Wert der Zahl PI
sin	Sinus
cos	Cosinus
tan	Tangens
atan	Arcustangens
exp	Exponentialfunktion zur Basis e
exp10	Exponentialfunktion zur Basis 10
log	Natürlicher Logarithmus
log10	Logarithmus zur Basis 10
wurzel	Quadratwurzel
fak	Fakultät

### Textfunktionen

links	Linke Teilkette bilden
rechts	Rechte Teilkette bilden
mitte	Teilkette bilden
klein	Kleinschreibung
gross	Großschreibung
text	Zahl in Zeichenkette wandeln

### Verschiedene Funktionen

attribute	Schreib- und Lesezugriff auf Zelle setzen
nachricht	Nachricht an Benutzer ausgeben

Fehlt eine Ihnen sehr wichtige Rechenfunktion? Schreiben Sie sie sich selbst! Wie das geht erfahren Sie im Abschnitt über Programmierung. Außerdem wird die Anzahl der eingebauten Rechenfunktionen bei neuen Versionen ständig aktualisiert.

## Rechenfunktionen in alphabetischer Ordnung

### atan(x)

Funktionsstyp:	Numerisch
Argument:	x: Kommazahl

Berechnet den Arcustanges des Arguments x.

### attribute(bereich;schreiben;lesen)

Funktionsstyp:	Numerisch
Argumente:	bereich: Zellbereich
	schreiben: Ganzzahl
	lesen: Ganzzahl

Im Zellbereich *bereich* wird die Schreib- und Leseerlaubnis gemäß den Werten der Argumente *schreiben* und *lesen* gesetzt. Folgende Werte sind für diese Parameter möglich:

1	Schreiben/Lesen verboten
0	Schreiben/Lesen erlaubt
-1	Schreib-/Leseerlaubnis bleibt unverändert

Der Funktionswert von *attribute* ist ohne Bedeutung.

#### Beispiel:

[a10] =attribute(a1:d4;1;-1)

Im Zellbereich a1:d4 wird das Beschreiben von Zellen verboten. Die Leseerlaubnis bleibt unverändert auf dem bestehenden Wert.

### cos(x)

Funktionsstyp:	Numerisch
Argument:	x: Kommazahl

Berechnet den Cosinus des Arguments x. x muß in Radian angegeben werden (2\*PI Radian = 360 Grad).

### datum(x)

Funktionsstyp:	Numerisch (Datum)
Argument:	x: Kommazahl

Formatiert den Wert der Zahl  $x$  als Datum.

### **exp(x)**

Funktionstyp: Numerisch  
Argument: x: Kommazahl

Berechnet die Exponentialzahl des Arguments  $x$  zur Basis  $e$ .

### **exp10(x)**

Funktionstyp: Numerisch  
Argument: x: Kommazahl

Berechnet die Exponentialzahl des Arguments  $x$  zur Basis 10.

### **fak(x)**

Funktionstyp: Numerisch  
Argument: x: Kommazahl

Berechnet die Fakultät des Arguments  $x$ .

### **gross(t)**

Funktionstyp: Text  
Argument: t: Zeichenkette

Wandelt die Buchstaben in der Zeichenkette  $t$  in Grossbuchstaben um.

### **heute()**

Funktionstyp: Numerisch (Datum)

Liefert den Wert des aktuellen Datums

### **jahr(x)**

Funktionstyp: Numerisch  
Argument: x: Numerisch (Datum)

Liefert die Jahreszahl des Datumswerts  $x$ .

### **jetzt()**

Funktionstyp: Numerisch (Datum)

Liefert den Wert der aktuellen Zeit.

### **klein(t)**

Funktionstyp: Text  
Argument: t: Zeichenkette

Wandelt die Buchstaben in der Zeichenkette  $t$  in Kleinbuchstaben um.

### **links(t;anzahl)**

Funktionstyp: Text  
Argumente: t: Zeichenkette  
anzahl: Ganzzahl

Löst aus der Zeichenkette  $t$  die linke Teilkette mit *anzahl* Zeichen aus.

### **log(x)**

Funktionstyp: Numerisch  
Argument: x: Kommazahl

Berechnet den Natürlichen Logarithmus des Arguments  $x$ .

### **log10(x)**

Funktionstyp: Numerisch  
Argument: x: Kommazahl

Berechnet den Logarithmus des Arguments  $x$  zur Basis 10.

### **max(r)**

Funktionstyp: Numerisch  
Argument: r: Bereich

Liefert den maximalen Wert der Zellen im Bereich  $r$ .

### **min(r)**

Funktionstyp: Numerisch  
Argument: r: Bereich

Liefert den minimalen Wert der Zellen im Bereich  $r$ .

### **minute(x)**

Funktionstyp: Numerisch  
Argument: x: Numerisch (Zeit)

Liefert die Minutenzahl des Zeitwerts  $x$ .

### **mitte(t;start;anzahl)**

Funktionstyp: Text  
Argumente: t: Zeichenkette  
start: Ganzzahl  
anzahl: Ganzzahl

Löst aus der Zeichenkette  $t$  eine Teilkette mit  $anzahl$  Zeichen aus, beginnend ab Zeichen  $start$ .

### **monat(x)**

Funktionstyp: Numerisch  
Argument: x: Numerisch (Datum)

Liefert die Monatszahl des Datumswerts  $x$ .

### **nachricht(t1;t2;knöpfe)**

Funktionstyp: Numerisch  
Argumente: t1: Zeichenkette  
t2: Zeichenkette  
knöpfe: Ganzzahl

Gibt eine Nachricht auf dem Bildschirm aus.  $t1$  ist die eigentliche Nachricht,  $t2$  der Titel der Dialogbox.  $knöpfe$  legt fest, welche Schaltflächen oder Bedienknöpfe in der Dialogbox erscheinen:

- 0 OK
- 1 OK Abbrechen
- 2 Abbrechen Wiederholen Ignorieren
- 3 Ja Nein Abbrechen
- 4 Ja Nein
- 5 Wiederholen Abbrechen

Der Funktionswert von *nachricht* zeigt an, mit welchem Knopf die Dialogbox geschlossen wurde:

- 1 OK
- 2 Abbrechen
- 3 Abbrechen
- 4 Wiederholen
- 5 Ignorieren
- 6 Ja
- 7 Nein

### **pi()**

Funktionstyp: Numerisch

Diese Funktion liefert den Wert der mathematischen Konstanten  $\pi$ .

### **rechts(t;anzahl)**

Funktionstyp: Text  
Argumente: t: Zeichenkette  
anzahl: Ganzzahl

Löst aus der Zeichenkette *t* die rechte Teilkette mit *anzahl* Zeichen aus.

### **sin(x)**

Funktionstyp: Numerisch  
Argument: x: Kommazahl

Berechnet den Sinus des Arguments *x*. *x* muß in Radian angegeben werden (2\*PI Radian = 360 Grad).

### **stunde(x)**

Funktionstyp: Numerisch  
Argument: x: Numerisch (Zeit)

Liefert die Stundenzahl des Zeitwerts *x*.

### **summe(r)**

Funktionstyp: Numerisch  
Argument: r: Bereich

Liefert die Summe der Werte der Zellen im Bereich *r*.

### **tag(x)**

Funktionstyp: Numerisch  
Argument: x: Numerisch (Datum)

Liefert die Tageszahl des Datumswerts *x*.

### **tan(x)**

Funktionstyp: Numerisch  
Argument: x: Kommazahl

Berechnet den Tangens des Arguments *x*. *x* muß in Radian angegeben werden (2\*PI Radian = 360 Grad).

### **text(x;stellen;dez)**

Funktionstyp: Text  
Argument: x: Kommazahl

stellen: Ganzzahl  
dez: Ganzzahl

Stellt den Wert des Arguments  $x$  als Zeichenkette dar. Die Zeichenkette ist *stellen* Zeichen lang, *dez* Nachkommastellen werden angegeben.

### **wurzel(x)**

Funktionstyp: Text  
Argument: x: Kommazahl

Berechnet die quadratische Wurzel des Arguments  $x$ .  $x$  muß positiv sein.

### **zeit(x)**

Funktionstyp: Numerisch (Zeit)  
Argument: x: Kommazahl

Formatiert den Wert der Zahl  $x$  als Zeit.

## Teil 3: Die Menufunktionen

### **Datei**

#### Neu

Das Arbeitsblatt wird initialisiert. Wurde das Arbeitsblatt zuvor geändert, wird die Speicherung der Daten angeboten. Wird hierbei die Option 'Abbrechen' gewählt, wird die Funktion Neu nicht durchgeführt. Andernfalls werden alle Zellen in den Grundzustand versetzt. In der Titelzeile des Adam-Hauptfensters wird [NEU] angezeigt.

#### Öffnen

Erlaubt das Öffnen eines Arbeitsblatts. Wurde das Arbeitsblatt zuvor geändert, wird die Speicherung der Daten angeboten. Wird hierbei die Option 'Abbrechen' gewählt, wird die Funktion Öffnen nicht durchgeführt.

Zur Auswahl der zu öffnenden Datei erscheint eine Dateiauswahlbox. Wird eine Datei ausgewählt so wird sie geladen und ihr Name erscheint in rechteckigen Klammern in der Titelzeile des Adam-Hauptfensters. Wird die Auswahl dagegen abgebrochen, so bleibt der alte Inhalt des Arbeitsblatts erhalten.

#### Speichern

Das Arbeitsblatt wird in der Datei gespeichert, aus der es geladen wurde. Handelt es sich um ein mit der Funktion Neu neu erzeugtes Arbeitsblatt, wird wie bei Speichern als verfahren.

#### Speichern als

Das Arbeitsblatt wird in eine Datei gespeichert, dessen Name durch eine Dateiauswahlbox

festgelegt wird. Enthält der Name keine Extension, wird diese mit '.adm' besetzt. Das Arbeitsblatt wird nun unter dem ausgewählten Namen geführt, dieser erscheint in der Titelzeile des Hauptfensters.

### Info

Zeigt Informationen über **Adam** und den Copyrightvermerk.

### Beenden

Die Arbeit mit **Adam** wird beendet. Wurde das Arbeitsblatt zuvor geändert, wird die Speicherung der Daten angeboten. Wird hierbei die Option 'Abbrechen' gewählt, wird die Funktion Beenden nicht durchgeführt.

Beim nächsten Start wird die zuletzt bearbeitete Datei automatisch geöffnet.

### **Bearbeiten**

Die Funktionen dieses Menu wirken auf die selktierten Zellen. Manche sind nur anwählbar wenn mehr als eine Zelle selektiert ist.

Die Funktionen Ausschneiden, Kopieren und Einfügen arbeiten mit einer Adam-spezifischen Ablage, in der Zellen zwischengespeichert werden.

### Ausschneiden

Die selektierten Zellen werden in die Ablage befördert und aus dem Arbeitsblatt entfernt. Etwa bereits abgelegte Daten gehen dabei verloren.

### Kopieren

Von den selektierten Zellen wird eine Kopie in der Ablage abgelegt. Etwa bereits abgelegte Daten gehen dabei verloren.

### Einfügen

Falls sich Zellen in der Ablage befindet werden sie beginned bei der aktiven Zelle in das Arbeitsblatt eingefügt. Der Inhalt der Ablage bleibt erhalten.

### Radieren

Der Inhalt der selektierten Zellen wird gelöscht.

### Vervielfachen

Der Inhalt der aktiven Zelle wird in die anderen selektierten Zellen übertragen.

### Schreibschutz

Der Schreibschutz für die selektierten Zellen wird ein- oder ausgeschaltet. In unter Schreibschutz gestellte Zellen kann nichts eingegeben werden.

### Bereichsname

Erlaubt die Zuordnung eines Bereichsnamens zu einem Zellbereichs. Im Eingabefeld der

Dialogbox ist zu Beginn des Dialogs bereits die Bezeichnung des selektierten Zellbereichs eingetragen. Links vom Gleichheitszeichen kann nun ein Name für den Bereich eingetragen werden. Es kann aber auch eine beliebige Definition eingetragen werden, unabhängig von der Vorgabe.

Außerdem können durch Anklicken einer Zeile in der Listbox unter dem Eingabefeld bereits vorhandene Definitionen in das Eingabefeld geholt und editiert werden.

Nicht mehr benötigte Bereichsnamen können gelöscht werden.

### Verbinden

Verbindet die selektierten Zellen einer Reihe zu einer spaltenübergreifenden *Verbundzelle*. Sind Zellen in mehreren Reihen selektiert, so wird in jeder Reihe eine Verbundzelle eingerichtet.

### Verbindung lösen

Funktioniert nur, wenn die aktuelle Zelle eine Verbundzelle ist. Die Verbundzelle wird in ihre Bestandteile aufgelöst.

### Zeile

Funktioniert nur, wenn eine oder mehrere komplette Zeilen selektiert sind (siehe Abschnitt 'Selektion'). Erlaubt das Einfügen und Löschen von Zeilen.

Die Operation findet statt in der Zeile mit der aktuellen Zelle. Alle Formeln des Arbeitsblatts werden in den Zelladressen aufbereitet, sowohl absolute als auch relative Adressierung.

### Spalte

Funktioniert nur, wenn eine oder mehrere komplette Spalten selektiert sind (siehe Abschnitt 'Selektion'). Erlaubt das Einfügen, Löschen und Formatieren von Spalten.

Einfügen und Löschen findet statt in Spalte Zeile mit der aktuellen Zelle. Alle Formeln des Arbeitsblatts werden in den Zelladressen aufbereitet, sowohl absolute als auch relative Adressierung.

Format erlaubt Einstellungen zum Format der Spalte. Spaltenbreite und Nachkommastellen für numerische Felder können eingestellt werden ebenso wie die Ausrichtung des externen Zelleninhalts bezüglich der Spalte. Die Ausrichtung wirkt allerdings nur auf Zellen vom Type Text, Datum und Zeit. Zahlen werden immer rechtsbündig ausgerichtet.

## **Funktion**

Diese Menufunktionen ermöglichen das Editieren von *benutzerdefinierten Rechenfunktionen*. Zur eigentlichen Programmierung der Rechenfunktionen finden Sie Informationen im Teil 'Programmierung'.

### Auswählen

Die zu bearbeitenden Rechenfunktion wird ausgewählt. Sie können über das Editierfeld einen neuen Namen eingeben oder durch Anklicken eines Namens in der Listbox eine vorhandene Rechenfunktion auswählen.

## Bearbeiten

Das Programmierfenster zum Editieren von Programmen und benutzerdefinierten Rechenfunktionen wird mit der aktuell bearbeiteten Rechenfunktion geöffnet. Sollte keine Funktion ausgewählt sein, wird zunächst wie in Auswählen verfahren.

## Löschen

Eine Rechenfunktion kann ausgewählt und gelöscht werden. Falls die Funktion noch benutzt wird, wird die Löschung unterbunden.

## **Rechnen**

Diese Menufunktion berechnet die Werte aller Zellen neu. Da **Adam** Zellen automatisch neu berechnet ist diese Funktion normalerweise ohne Wirkung. Es kann jedoch sein, daß die automatische Neuberechnung nicht möglich ist (siehe Abschnitt 'Automatische Berechnung').

Wirkung hat diese Funktion auch, wenn man eine *benutzerdefinierte Rechenfunktion* ändert welche nicht direkt durch eine Formel oder ein Programm in einer Zelle benutzt wird, sondern durch eine andere *benutzerdefinierte Rechenfunktion* aufgerufen wird. In diesem Fall kann keine automatische Neuberechnung stattfinden, weil keine Querverweise zu einer Zelle vorliegen.

## **Einstellung**

### Format global

Hier kann die Spaltenbreite aller Spalten eingestellt werden. Es kann festgelegt werden, ob die Zellen durch ein Punkt- oder Linienraster unterteilt werden.

### Autosicherung

Hier wird festgelegt, nach wieviel Tastaturanschlägen die automatische Datensicherung stattfindet. Ein Wert von 0 schaltet sie ab. Der Knopf 'aus' setzt den Wert auf 0.

## **Austausch**

### Drucken

Das Arbeitsblatt wird auf dem Drucker ausgegeben. Da es im allgemeinen nicht auf eine Seite passt, kann man *Randreihen* und *-spalten* einstellen. Das sind die ersten Reihen und Spalten des Arbeitsblatts die auf jeder Seite wiederholt ausgegeben werden sollen um eine Orientierung beim Lesen zu geben. Außerdem kann man einstellen, ob die Reihen- und Spaltennamen (*Titel*) mitgedruckt werden sollen.

### Datenbank

Mit dieser Menufunktion können Datensätze aus dem Datenbankprogramm 'Vesuv' in die Tabellenkalkulation eingelesen werden.

Die Datensätze werden ab der Reihe mit der aktiven Zelle in das Arbeitsblatt gespeichert. Jedes Feld des Datensatzes belegt eine Zelle.

Gelesen werden die aktuell selektierten Datensätze der Datenbank. Wurde keine Selektion in

'Vesuv' gemacht so sind das alle Datensätze!

## Grafik

Diese Menufunktion erlaubt die Übertragung ausgesuchter Zellen in das Grafikprogramm 'Albrecht' zur grafischen Darstellung mit diversen Diagrammen.

## Teil 4: Programmierung

Eines der herausragendsten Kennzeichen von Adam ist seine Programmierbarkeit. Jede Zelle kann nicht nur eine Formel sondern ein komplettes Programm enthalten. Außerdem können *benutzerdefinierte Rechenfunktionen* auf diesem Weg erstellt werden, welche genau wie eingebaute Funktionen in Formeln und Programmen verwendet werden können.

Will man die Programmierfähigkeit von Adam benutzen sollte man Grundkenntnisse in der Programmierung allgemein besitzen, vielleicht erworben in Basic oder Pascal. Falls Sie sich noch nie mit Programmierung beschäftigt haben sollten Sie nicht hier damit beginnen, dazu ist diese Dokumentation nicht hinreichend!

Eine Zelle wird zur Programmzelle, indem man ein @ in sie eingibt. Dieses Zeichen kann nicht überschrieben werden, da man damit ein eventuell größeres Programm zerstören kann. Programmzellen müssen explizit mit \Bearbeiten\Radieren entfernt werden.

Bei einer Eingabe von `Enter` in eine Programmzelle oder einen Doppelklick mit der linken Maustaste wird ein kleines Editorfenster geöffnet in dem das Programm bearbeitet werden kann. Bei weiterer Eingabe von `Enter` wird das Fenster wieder geschlossen, das Programm wird übersetzt und bei Fehlerfreiheit ausgeführt. Ist ein Fehler enthalten wird eine Fehlermeldung ausgegeben und das Fenster wieder geöffnet. (Tip: sollten Sie den Fehler nicht beheben können so kann die betreffende Zeile durch Voranstellen von 'rem' in eine Kommentarzeile verwandelt werden die nicht ausgeführt wird - siehe *Anweisung* 'rem'). Der Titel der Fehlermeldung enthält den Namen der Zelle und die Nummer der fehlerhaften Zeile, nach einem Doppelpunkt folgt das Symbol welches den Fehler verursacht.

Bei Eingabe von `ESC` wird der ursprüngliche Inhalt bei Öffnen des Fensters wiederhergestellt.

Statt `Enter` oder `ESC` können auch die Knöpfe 'OK' bzw. 'Rück' am oberen Rand des Editorfensters benutzt werden.

Das Ende einer Zeile im Programm wird durch `Ctrl+Enter` herbeigeführt.

*Benutzerdefinierte Rechenfunktionen* werden auf dieselbe Weise bearbeitet, nur das die zu bearbeitende Funktion über das Menu 'Funktion' ausgewählt wird. Außerdem beginnen Funktionen stets mit der *Anweisung* 'funktion' und enden mit der *Anweisung* 'endfun'.

Wird eine Rechenfunktion geändert, so werden Zellen welche eine Formel oder eine Programm enthalten die diese Funktion benutzen automatisch neu berechnet. Dies kann etwas länger dauern als die Neuberechnung aufgrund der Änderung von Zellen. Wird diese Funktion nur von einer anderen Funktion aufgerufen findet aufgrund der fehlenden Querverweise keine automatische Neuberechnung statt. Hier muß die Menufunktion \Rechnen benutzt werden.

Ein Programm besteht aus einer oder mehreren Zeilen mit *Anweisungen*. Anweisungen die Sie schon kennen sind Formeln. Auch in Programmen können Sie Formeln benutzen. Anders als in

gewöhnlichen Zellformeln können diese nicht nur die eigene Zelle beeinflussen sondern jede Zelle, in dem man deren Adresse vor das Gleichheitszeichen setzt in der Art:

a1=.....

In den benutzerdefinierten Rechenfunktionen dürfen allerdings keine Zelladressen benutzt werden. Solche Funktion dürfen nur über das Liefern des Funktionswertes nach außen wirken.

### Freie Variablen

Beim Programmieren sind sogenannte *freie Variablen* von Interesse. Das sind Adressen von Speicher abseits der Zellen des Arbeitsblatts. Freie Variable gibt es als Kommazahlen und als Ganzzahlen. Ihre Namen sind beliebig, das erste Zeichen muß jedoch ein Buchstabe sein, und er darf keine Zellbezeichnung sein. Die Namen der Ganzzahlen haben ein %-Zeichen angehängt. Beispiele für Namen freier Variablen:

a, summe, maximum	Kommazahlvariablen
n%, anzahl%, fmin%	Ganzzahlvariablen

Freie Variablen müssen nicht deklariert oder vereinbart werden. Sie werden bei Ihrer ersten Nennung erzeugt. Werden in verschiedenen Programmen die gleichen Variablen benutzt, so belegen sie verschiedene Speicherstellen.

### Programmabbruch erzwingen

Ein häufiger Fehler bei der Erstellung von Programmen ist die versehentliche Programmierung einer Endlosschleife, das Programm kommt nicht zum Halten. In diesem Fall kann es durch die Taste `Ctrl-a` abgebrochen werden.

### Übersicht der Anweisungen nach Bereichen

rem           Kommentarzeile

#### Von-Schleife

von	Von-Schleife einleiten
endvon	Von-Schleife abschliessen

#### Solange-Schleife

solange	Solange-Schleife einleiten
endsol	Solange-Schleife abschliessen

#### Bereich

bereich	Bereich einleiten
endber	Bereich abschliessen

#### Wenn-Abfrage

wenn	Wenn-Abfrage beginnen
sonstwenn	Alternative prüfen
sonst	Alternative
endwenn	Ende Wenn-Abfrage

## Abbruch

abbruch            Schleife abbrechen

## Funktion

funktion            Funktion einleiten  
endfun              Funktion beenden  
fwert                Funktionswert setzen

## **Anweisungen**

### **rem**

Leitet eine Kommentarzeile ein. Diese Zeile wird weder übersetzt noch ausgeführt.

**von** var%=start **auf** ende [**schritt** delta]

...

**endvon**

Diese Gruppe von Anweisungen definiert eine Schleife, d. h. die Anweisungen zwischen 'von' und 'endvon' werden wiederholt durchlaufen, abhängig vom Wert der Variablen *var%*. *var%* ist immer *freie Ganzzahlvariable*.

Die großen *start*, *ende* und *schritt* können Variablen (Zellen oder freie) oder Konstanten sein, sie sollten aber möglichst Ganzzahlkonstanten oder -variablen sein, andernfalls werden sie automatisch gewandelt. Beim Eintritt in die Von-Schleife wird die Variable *var%* auf den Wert *start* gesetzt. Sie wird bei jedem Erreichen der Anweisung 'endvon' um den Wert *delta* erhöht.

Falls der Teil der Anweisung 'von' in den rechteckigen Klammern fehlt, wird delta mit '1' angenommen. Falls beim Erreichen der Anweisung 'endvon' der Wert von *var%* größer ist als *ende* wird die Von-Schleife verlassen und die nächste Anweisung nach 'endvon' ausgeführt.

Beispiel: Berechnen der Fakultät einer Zahl (falls Sie nicht die eingebaute Funktion vorziehen)

(Programm in Zelle b2)

```
a2=1,0
von n%=2 auf a1
  a2=a2*n%
endvon
```

In Zelle a2 steht dann der Wert der Fakultät der Zahl in Zelle a1. Das Programm kann in jeder Zelle des Arbeitsblatts stehen, da die Formel in der dritten Zeile explizit die Zelle a2 mit einem Wert besetzt. Schreibt man das Programm in der Form

```
a2=1,0
von n%=2 auf a1
  =a2*n%
endvon
```

muß es in Zelle a2 stehen, da die Formel in Zeile 3 jetzt die eigene Zelle besetzt.

Soll die Schleife nicht aufwärts gezählt werden, d. h. *ende* ist größer als *start*, sondern abwärts, d. h. *ende* ist kleiner als *start*, muß statt des Worts 'auf' da Wort 'ab' gesetzt werden. dann wird bei Erreichen von 'endvon' *delta* von *var%* subtrahiert.

**solange** vergleich

...

**endsol**

Diese beiden Anweisungen bilden eine Solange-Schleife. Diese Schleife wird solange durchlaufen wie der Vergleichsausdruck *vergleich* wahr ist. Dann wird die Schleife verlassen und die nächste Anweisung nach 'endsol' ausgeführt. Ist der Vergleichsausdruck zu Beginn bereits nicht erfüllt, wird die Schleife nicht durchlaufen und gleich nach der nächsten Anweisung hinter 'endsol' gesprungen.

Der Vergleichsausdruck *vergleich* besteht aus zwei arithmetischen Ausdrücken (oder 'Formelausdrücken') die durch einen Vergleichsoperator miteinander verglichen werden. Vergleichsoperatoren sind:

=	gleich
>	größer als
>=	größer oder gleich
<	kleiner als
<=	kleiner oder gleich
<>	ungleich

Beispiele für Vergleichsausdrücke:

```
a1 > a2+10,0  
a1+a2 <= a3  
sin(a1) = 1,0
```

Solche Vergleichsausdrücke können durch die *logischen* Operatoren 'und' und 'oder' kombiniert werden:

```
a1 < a2 und a2 < 11,0  
a1+a2 > wurzel(2,0) oder a3 = 0
```

**bereich** zellbereich

...

**endber**

Die zwischen den Anweisungen 'bereich' und 'endber' liegenden Anweisungen werden für jede Zelle im Bereich *zellbereich* ausgeführt. Zur Definition von Zellbereichen lesen Sie bitte den Abschnitt 'Bereiche' im Teil über Rechenfunktionen und über die Menufunktion \Bearbeiten\ Bereichsname im Teil über die Menufunktionen.

Um beim Durchlaufen der Bereichs-Schleife auf die jeweils bearbeitete Zelle zugreifen zu können gibt es den *Stellvertreter* @. Er repräsentiert den Zellnamen und kann wie dieser eingesetzt werden. Beispiele:

```
(Programm in Zelle b4)  
c6=0,0  
bereich a1:b2  
c6=c6+@
```

endber

ist eine Art Summenfunktion. In Zelle c6 wird die Summe des Bereichs a1:b2 gebildet. Dazu wird beim Durchlauf der Bereichs-Schleife durch den Bereich a1:b2 die jeweils bearbeitete Zelle @ zu c6 aufaddiert.

(Programm in Zelle a1)

```
n%=1
bereich a2:f2
@=n%
n%=n%+1
endber
```

Die Ganzzahlvariable n% wird beim Durchlaufen der Bereichs-Schleife im Bereich a2:f2 hochgezählt. Der jeweils bestehende Wert wird in bearbeitete Zelle @ des Bereichs gespeichert.

**wenn** vergleich **dann** anweisung

Das ist eine einfache Wenn-Abfrage. Wenn der Vergleichsausdruck (siehe Anweisung 'solnage') wahr ist wird die Anweisung *anweisung* durchgeführt, ansonsten wird nichts getan. *anweisung* kann jede beliebige einfache Anweisung sein, Formeln z.B. oder die Anweisung 'abbruch', aber keine Schleifen wie 'von', 'solange' usw..

**wenn** vergleich **dann**

```
...
[sonstwenn vergleich1 ('swenn' Abkürzung für 'sonst wenn')
...]
[sonst
...]
endwenn
```

Dies ist die strukturierte Wenn-Abfrage. Nach 'dann' folgt keine weitere Angabe, dafür muß die Abfrage mit einer Anweisung 'end wenn' abgeschlossen werden. Alle Zeilen die auf 'wenn' folgen werden bei wahren *vergleich* ausgeführt. Die Zweige 'sonst wenn' und 'sonst' sind optional und können bei Bedarf benutzt werden.

Ist der Vergleich *vergleich* unwahr, so führt 'sonst wenn' führt einen weiteren Vergleich *vergleich1* durch, die darauffolgenden Anweisungen werden bei wahren Vergleich durchgeführt. Es können mehrere 'sonst wenn'-Anweisungen mit beliebigen Vergleichen folgen.

Die Anweisung 'sonst' führt keinen Vergleich durch. Die auf sie folgenden Anweisungen werden ausgeführt falls keine Bedingung vorher erfüllt war. 'sonst' kann nicht vor 'sonst wenn' erscheinen.

Beispiele:

(Programm in Zelle b6)

```
wenn a1=1,0 dann
a2=10,0
nachricht("a1=1";"Message";0)
end wenn
```

```
wenn a1=1,0 dann
```

```
a2=10,0
sonst
a2=12,2
endwenn
```

```
wenn a1=1,0 dann
a2=10,0
sonstwenn a1=2,0
a2=10,2
sonstwenn a1=3,0
a2=10,2
sonst
nachricht("falscher wert";"a1";0)
endwenn
```

### **abbruch**

Diese Anweisung dient dazu die Schleifen 'von', 'solange' und 'bereich' abubrechen. Sie wird im allgemeinen zusammen mit der Anweisung 'wenn' benutzt um bei Eintreten einer bestimmten Bedingung die Schleife vor dem regulären Ende zu verlassen.

#### Beispiel:

```
(Programm in Zelle b2)
a2=0,0
von i%=1 auf a1
a2=a2*i%
wenn a2 >= 1D6 dann
nachricht("Zahl zu groß";"a2";0)
abbruch
enwenn
endvon
```

Eine ähnliche Fakultätsfunktion hatten wir bereits bei Erläuterung der Anweisung 'von'. Hier wird eine 'Bremse' eingebaut, falls die Zahl zu groß wird.

**funktion** fname([[par1;]par2...])

...

**fwert** wert

...

**endfun**

Durch die Anweisung 'funktion' wird eine *benutzerdefinierte Rechenfunktion* eingeleitet, mit 'endfun' wird sie beendet. Wenn Sie in der Menufunktion \Funktion\Auswählen einen neuen Funktionsnamen eingeben, werden diese beiden Anweisungen automatisch generiert. Sie müssen allerdings noch in den Klammern die Funktionsparameter eintragen, falls die Funktion Parameter haben soll; sie werden durch Semikolons getrennt. Eine Funktion muß aber keine Parameter haben.

Im allgemeinen wird die Funktion einen Funktionswert zurückliefern sollen. Dieser wird durch die Anweisung 'fwert' festgelegt. 'fwert' ist i.a. die letzte Anweisung vor 'endfun', dies muß aber nicht sein.

#### Beispiel:

```
funktion cos(x)
sinx=sin(x)
wert wurzel(1,0-sinx*sinx)
endfun
```