

Version 10 Release 1  
Oktober 2017

*Benutzerhandbuch zu IBM-Makros für  
IBM Marketing Software*

**IBM**

**Hinweis**

Vor Verwendung dieser Informationen und des darin beschriebenen Produkts sollten die Informationen unter „Bemerkungen“ auf Seite 167 gelesen werden.

Diese Ausgabe bezieht sich auf Version 10, Release 1, Modifikation 0 von IBM Campaign und alle nachfolgenden Releases und Modifikationen, bis dieser Hinweis in einer Neuauflage geändert wird.

© Copyright IBM Corporation 1998, 2017.

---

# Inhaltsverzeichnis

## Kapitel 1. Verwendung von Makros in IBM Campaign . . . . . 1

Zusammenfassung der Makrofunktionen für IBM Campaign . . . . .	1
Statistische Funktionen . . . . .	1
Mathematische und trigonometrische Funktionen . . . . .	2
Zeichenfolgefunktionen . . . . .	6
Datums- und Zeitfunktionen . . . . .	7
Gruppierungsfunktionen . . . . .	8
Verschiedene Funktionen . . . . .	8
Makrofunktionsparameter für IBM Campaign . . . . .	9
Formatspezifikationen . . . . .	9
Verwendung von Konstanten . . . . .	9

## Kapitel 2. Verwendung von Makros in IBM Interact . . . . . 11

Zusammenfassung der Makrofunktionen für IBM Interact . . . . .	11
Statistische Funktionen . . . . .	11
Mathematische und trigonometrische Funktionen . . . . .	12
Zeichenfolgefunktionen . . . . .	13
Datums- und Zeitfunktionen . . . . .	14
Verschiedene Funktionen . . . . .	14
Makrofunktionsparameter für IBM Interact . . . . .	15
Formatspezifikationen . . . . .	15
Verwendung von Konstanten . . . . .	15

## Kapitel 3. IBM Makroreferenz . . . . . 17

Gültige Datumsformatschlüsselwörter . . . . .	17
ABS-Makro . . . . .	19
ACOS-Makro . . . . .	20
ACOT-Makro . . . . .	21
ADD_MONTHS-Makro . . . . .	23
AND-Makro . . . . .	24
ASIN-Makro . . . . .	26
ATAN-Makro . . . . .	28
AVG-Makro . . . . .	29
BETWEEN-Makro . . . . .	31
BIT_AND-Makro . . . . .	32
BIT_NOT-Makro . . . . .	33
BIT_OR-Makro . . . . .	35
BIT_XOR-Makro . . . . .	37
CEILING-Makro . . . . .	38
COLUMN-Makro . . . . .	39
COS-Makro . . . . .	41
COSH-Makro . . . . .	42
COT-Makro . . . . .	44
COUNT-Makro . . . . .	45
CURRENT_DATE-Makro . . . . .	47
CURRENT_DAY-Makro . . . . .	48
CURRENT_JULIAN-Makro . . . . .	48
CURRENT_MONTH-Makro . . . . .	49
CURRENT_TIME-Makro . . . . .	50
Datumseinstellung in Ihrer Webanwendung . . . . .	50
CURRENT_WEEKDAY-Makro . . . . .	51

CURRENT_YEAR-Makro . . . . .	52
DATE . . . . .	52
DATE_FORMAT-Makro . . . . .	54
DATE_JULIAN-Makro . . . . .	56
DATE_STRING-Makro . . . . .	56
DAY_BETWEEN-Makro . . . . .	58
DAY_FROMNOW-Makro . . . . .	59
DAY_INTERVAL-Makro . . . . .	60
DAYOF-Makro . . . . .	61
DISTANCE-Makro . . . . .	61
DIV-Makro . . . . .	62
EQ-Makro . . . . .	64
EXP-Makro . . . . .	66
EXTERNALCALLOUT-Makro . . . . .	67
FACTORIAL-Makro . . . . .	68
FLOOR-Makro . . . . .	69
FORMAT-Makro . . . . .	70
FRACTION-Makro . . . . .	73
GE-Makro . . . . .	74
GROUPBY-Makro . . . . .	76
GROUPBY_WHERE-Makro . . . . .	79
GT-Makro . . . . .	80
IF-Makro . . . . .	82
IN-Makro . . . . .	84
INT-Makro . . . . .	84
INVERSE-Makro . . . . .	86
IS-Makro . . . . .	87
ISERROR-Makro . . . . .	87
ISODD-Makro . . . . .	88
ISEVEN-Makro . . . . .	89
ISODD-Makro . . . . .	90
LE-Makro . . . . .	91
LIKE-Makro . . . . .	93
LN- oder LOG-Makro . . . . .	95
LOG2-Makro . . . . .	96
LOG10-Makro . . . . .	98
LOWER-Makro . . . . .	99
LT-Makro . . . . .	100
LTRIM-Makro . . . . .	101
MAX-Makro . . . . .	102
MEAN-Makro . . . . .	103
MIN-Makro . . . . .	105
MINUS-Makro . . . . .	107
MOD-Makro . . . . .	108
MONTHOF-Makro . . . . .	110
MULT-Makro . . . . .	111
NE-Makro . . . . .	113
NOT-Makro . . . . .	115
NUMBER-Makro . . . . .	116
OR-Makro . . . . .	122
POSITION-Makro . . . . .	124
PLUS-Makro . . . . .	125
POW-Makro . . . . .	127
RANDOM-Makro . . . . .	128
RANDOM_GAUSS-Makro . . . . .	130
ROUND-Makro . . . . .	131

ROWNUM-Makro . . . . .	132
RTRIM-Makro . . . . .	133
SIGN-Makro . . . . .	133
SIN-Makro . . . . .	134
SINH-Makro . . . . .	135
SQRT-Makro . . . . .	137
STDV- oder STDEV-Makro . . . . .	138
STRING_CONCAT-Makro . . . . .	140
STRING_HEAD-Makro . . . . .	141
STRING_LENGTH-Makro . . . . .	143
STRING_PROPER-Makro . . . . .	144
STRING_SEG-Makro . . . . .	144
STRING_TAIL-Makro . . . . .	146
SUBSTR- oder SUBSTRING-Makro . . . . .	147
SUM-Makro . . . . .	148
TAN-Makro . . . . .	150
TANH-Makro . . . . .	151

TOTAL-Makro . . . . .	153
TRUNCATE-Makro . . . . .	154
UPPER-Makro . . . . .	156
VARIANCE-Makro . . . . .	156
WEEKDAY-Makro . . . . .	158
WEEKDAYOF-Makro . . . . .	159
XOR-Makro . . . . .	160
YEAROF-Makro . . . . .	162

**Bevor Sie sich an den technischen Support von IBM wenden . . . . . 165**

**Bemerkungen . . . . . 167**

Marken . . . . .	169
Hinweise zu Datenschutzrichtlinien und Nutzungsbedingungen . . . . .	169

---

# Kapitel 1. Verwendung von Makros in IBM Campaign

Dieses Kapitel enthält Informationen zur Verwendung von Makros in IBM® Campaign. Lesen Sie dieses Kapitel unbedingt, bevor Sie versuchen, Makros in IBM Campaign zu verwenden.

---

## Zusammenfassung der Makrofunktionen für IBM Campaign

Die Tabellen in diesem Abschnitt fassen die Makrofunktionen nach Kategorien zusammen.

Ausführliche Referenzseiten zu den einzelnen Makrofunktionen in alphabetischer Reihenfolge werden in Kapitel 3, „IBM Makroreferenz“, auf Seite 17 bereitgestellt.

**Wichtig:** Makros können für IBM Campaign und IBM Interact oder nur für eines der beiden Produkte Anwendung finden. In den Makrobeschreibungen werden die Produkte angegeben, in denen das jeweilige Makro verfügbar ist.

Weitere Informationen zu den Eingabeparametern der Makrofunktionen finden Sie unter „Makrofunktionsparameter für IBM Campaign“ auf Seite 9.

### Statistische Funktionen

Makroname	Rückgabe	Beschreibung
AVG	Bei dem Schlüsselwort ALL ein Einzelwert in einer neuen Spalte; bei dem Schlüsselwort COL eine Spalte mit einem Einzelwert für jede Eingabespalte; bei dem Schlüsselwort ROW eine Spalte mit einem Wert für jede Zeile	Berechnet das arithmetische Mittel oder den Durchschnitt eines Zellenbereichs
COUNT	Einzelwert in einer neuen Spalte	Zählt die Anzahl der Werte in einem angegebenen Datenbereich
MAX	Bei dem Schlüsselwort ALL ein Einzelwert in einer neuen Spalte; bei dem Schlüsselwort COL eine Spalte mit einem Einzelwert für jede Eingabespalte; bei dem Schlüsselwort ROW eine Spalte mit einem Wert für jede Zeile	Berechnet das Maximum eines Zellenbereichs
MEAN	Bei dem Schlüsselwort ALL ein Einzelwert in einer neuen Spalte; bei dem Schlüsselwort COL eine Spalte mit einem Einzelwert für jede Eingabespalte; bei dem Schlüsselwort ROW eine Spalte mit einem Wert für jede Zeile	Berechnet das arithmetische Mittel oder den Durchschnitt eines Zellenbereichs

<b>Makroname</b>	<b>Rückgabe</b>	<b>Beschreibung</b>
MIN	Bei dem Schlüsselwort ALL ein Einzelwert in einer neuen Spalte; bei dem Schlüsselwort COL eine Spalte mit einem Einzelwert für jede Eingabespalte; bei dem Schlüsselwort ROW eine Spalte mit einem Wert für jede Zeile	Berechnet das Minimum eines Zellenbereichs
STDV oder STDEV	Bei dem Schlüsselwort ALL ein Einzelwert in einer neuen Spalte; bei dem Schlüsselwort COL eine Spalte mit einem Einzelwert für jede Eingabespalte; bei dem Schlüsselwort ROW eine Spalte mit einem Wert für jede Zeile	Berechnet die Standardabweichung eines Zellenbereichs
VARIANCE	Bei dem Schlüsselwort ALL ein Einzelwert in einer neuen Spalte; bei dem Schlüsselwort COL eine Spalte mit einem Einzelwert für jede Eingabespalte; bei dem Schlüsselwort ROW eine Spalte mit einem Wert für jede Zeile	Berechnet die Varianz eines Zellenbereichs

## Mathematische und trigonometrische Funktionen

<b>Makroname</b>	<b>Rückgabe</b>	<b>Beschreibung</b>
ABS	Eine Spalte für jede Eingabespalte	Berechnet den absoluten Wert des Inhalts des angegebenen Datenbereichs.
ACOS	Eine Spalte für jede Eingabespalte	Berechnet den Arkuskosinus des Inhalts des angegebenen Datenbereichs.
ACOT	Eine Spalte für jede Eingabespalte	Berechnet den Arkuskotangens des Inhalts des angegebenen Datenbereichs.
ASIN	Eine Spalte für jede Eingabespalte	Berechnet den Arkussinus des Inhalts des angegebenen Datenbereichs.
ATAN	Eine Spalte für jede Eingabespalte	Berechnet den Arkustangens des Inhalts des angegebenen Datenbereichs.
AVG	Eine Spalte für jede Eingabespalte	Berechnet das arithmetische Mittel oder den Durchschnitt der Zellen im angegebenen Datenbereich.
BETWEEN	Eine Spalte für jede Eingabespalte	Vergleicht zwei Werte, um festzustellen, ob der angegebene Wert zwischen zwei anderen Werten liegt.

<b>Makroname</b>	<b>Rückgabe</b>	<b>Beschreibung</b>
CEILING	Eine Spalte für jede Eingabespalte	Berechnet die Obergrenze jedes Werts im angegebenen Datenbereich.
COLUMN	Eine Spalte für jede Eingabespalte	Erstellt neue Spalten, wobei die Eingabewerte in jeder Spalte vertikal verkettet werden.
COS	Eine Spalte für jede Eingabespalte	Berechnet den Kosinus des Inhalts des angegebenen Datenbereichs.
COSH	Eine Spalte für jede Eingabespalte	Berechnet den Hyperbelkosinus des Inhalts des angegebenen Datenbereichs.
COT	Eine Spalte für jede Eingabespalte	Berechnet den Kotangens des Inhalts des angegebenen Datenbereichs.
COUNT	Eine Spalte mit einem Einzelwert	Zählt die Anzahl der Zellen, die Werte enthalten, im angegebenen Datenbereich.
EXP	Eine Spalte für jede Eingabespalte	Erhebt die natürliche Zahl (e) in die Potenz, die durch den Inhalt jeder Zelle im angegebenen Datenbereich angegeben ist.
FACTORIAL	Eine Spalte für jede Eingabespalte	Berechnet die Fakultät jedes Werts im angegebenen Datenbereich.
FLOOR	Eine Spalte für jede Eingabespalte	Berechnet die Untergrenze jedes Werts im angegebenen Datenbereich.
FRACTION	Eine Spalte für jede Eingabespalte	Berechnet die Nachkommastellen jedes Werts im angegebenen Datenbereich.
INT	Eine Spalte für jede Eingabespalte	Berechnet den abgerundeten Ganzzahlwert des Inhalts des angegebenen Datenbereichs.
INVERSE	Eine Spalte für jede Eingabespalte	Berechnet den Negativwert des Inhalts des angegebenen Datenbereichs.
LN	Eine Spalte für jede Eingabespalte	Berechnet den natürlichen Logarithmus des Inhalts des angegebenen Datenbereichs.
LOG	Eine Spalte für jede Eingabespalte	Berechnet den natürlichen Logarithmus des Inhalts des angegebenen Datenbereichs.
LOG2	Eine Spalte für jede Eingabespalte	Berechnet den binären Logarithmus des Inhalts des angegebenen Datenbereichs.

<b>Makroname</b>	<b>Rückgabe</b>	<b>Beschreibung</b>
LOG10	Eine Spalte für jede Eingabespalte	Berechnet den dekadischen Logarithmus des Inhalts des angegebenen Datenbereichs.
MAX	Bei dem Schlüsselwort ALL ein Einzelwert in einer neuen Spalte; bei dem Schlüsselwort COL eine Spalte mit einem Einzelwert für jede Eingabespalte; bei dem Schlüsselwort ROW eine Spalte mit einem Wert für jede Zeile	Berechnet das Maximum eines Zellenbereichs
MEAN	Bei dem Schlüsselwort ALL ein Einzelwert in einer neuen Spalte; bei dem Schlüsselwort COL eine Spalte mit einem Einzelwert für jede Eingabespalte; bei dem Schlüsselwort ROW eine Spalte mit einem Wert für jede Zeile	Berechnet das arithmetische Mittel oder den Durchschnitt eines Zellenbereichs
MIN	Bei dem Schlüsselwort ALL ein Einzelwert in einer neuen Spalte; bei dem Schlüsselwort COL eine Spalte mit einem Einzelwert für jede Eingabespalte; bei dem Schlüsselwort ROW eine Spalte mit einem Wert für jede Zeile	Berechnet das Minimum eines Zellenbereichs
RANDOM	Eine Spalte mit der angegebenen Anzahl von Werten	Gibt die angegebene Anzahl von Zufallszahlen zurück.
RANDOM_GAUSS	Eine Spalte mit der angegebenen Anzahl von Werten	Gibt die angegebene Anzahl von Zufallswerten aus einer gaußschen Verteilung zurück.
ROUND	Eine Spalte für jede Eingabespalte	Berechnet den gerundeten Wert des Inhalts des angegebenen Datenbereichs.
SIGN	Eine Spalte für jede Eingabespalte	Berechnet das Vorzeichen (positiv oder negativ) der Werte im angegebenen Datenbereich.
SIN	Eine Spalte für jede Eingabespalte	Berechnet den Sinus des Inhalts des angegebenen Datenbereichs.
SINH	Eine Spalte für jede Eingabespalte	Berechnet den Hyperbelsinus des Inhalts des angegebenen Datenbereichs.
SQRT	Eine Spalte für jede Eingabespalte	Berechnet die Quadratwurzel des Inhalts des angegebenen Datenbereichs.

<b>Makroname</b>	<b>Rückgabe</b>	<b>Beschreibung</b>
STDV oder STDEV	Bei dem Schlüsselwort ALL ein Einzelwert in einer neuen Spalte; bei dem Schlüsselwort COL eine Spalte mit einem Einzelwert für jede Eingabespalte; bei dem Schlüsselwort ROW eine Spalte mit einem Wert für jede Zeile	Berechnet die Standardabweichung eines Zellenbereichs
SUM	Bei dem Schlüsselwort ALL ein Einzelwert in einer neuen Spalte; bei dem Schlüsselwort COL eine Spalte mit einem Einzelwert für jede Eingabespalte; bei dem Schlüsselwort ROW eine Spalte mit einem Wert für jede Zeile	Berechnet die Summe eines Zellenbereichs.
TAN	Eine Spalte für jede Eingabespalte	Berechnet den Tangens des Inhalts des angegebenen Datenbereichs.
TANH	Eine Spalte für jede Eingabespalte	Berechnet den Hyperbeltangens des Inhalts des angegebenen Datenbereichs.
TOTAL	Bei dem Schlüsselwort ALL ein Einzelwert in einer neuen Spalte; bei dem Schlüsselwort COL eine Spalte mit einem Einzelwert für jede Eingabespalte; bei dem Schlüsselwort ROW eine Spalte mit einem Wert für jede Zeile	Berechnet die Summe eines Zellenbereichs.
TRUNCATE	Eine Spalte für jede Eingabespalte	Berechnet den Ganzzahlanteil jedes Werts im angegebenen Datenbereich.
VARIANCE	Bei dem Schlüsselwort ALL ein Einzelwert in einer neuen Spalte; bei dem Schlüsselwort COL eine Spalte mit einem Einzelwert für jede Eingabespalte; bei dem Schlüsselwort ROW eine Spalte mit einem Wert für jede Zeile	Berechnet die Varianz eines Zellenbereichs

## Zeichenfolgefunktionen

Makroname	Rückgabe	Beschreibung
FORMAT	Eine Spalte für jede Eingabespalte	Steuert die Ausgabeformatierung für Zahlen und Zeichenfolgen (z. B. Ausgabebreite, Ausrichtung, numerische Genauigkeit, Dezimalzeichen, Trennzeichen usw.). Gibt die formatierte Ausgabezeichenfolge zurück.
LIKE	Eine Spalte für jede Eingabespalte	Ermittelt, ob eine Zeichenfolge mit einem angegebenen Muster übereinstimmt
LOWER	Eine Spalte für jede Eingabespalte	Konvertiert Zeichenfolgewerte in Kleinschreibung
LTRIM	Eine Spalte für jede Eingabespalte	Entfernt führende Leerzeichen aus jedem Zeichenfolgewart
NUMBER	Eine Spalte für jede Eingabespalte	Konvertiert ASCII-Zeichenfolgen für Uhrzeit- und Datumsangaben in numerische Werte
POSITION	Eine Spalte für jede Eingabespalte	Gibt die Anfangsposition eines Musters in einer Zeichenfolge zurück
RTRIM	Eine Spalte für jede Eingabespalte	Entfernt nachfolgende Leerzeichen aus jedem Zeichenfolgewart
STRING_CONCAT	Eine Spalte mit einem Wert für jede Zeile der kürzesten Eingabespalte	Verkettet Zeichenfolgen aus den angegebenen Datenbereichen
STRING_HEAD	Eine Spalte für jede Eingabespalte	Gibt die ersten <i>n</i> Zeichen jeder Zeichenfolge im angegebenen Datenbereich zurück
STRING_LENGTH	Eine Spalte für jede Eingabespalte	Gibt die Länge jeder Zeichenfolge im angegebenen Datenbereich zurück
STRING_PROPER	Eine Spalte für jede Eingabespalte	Konvertiert jeden Zeichenfolgewart so, dass der erste Buchstabe oder irgendein Buchstabe, der auf ein Leerzeichen oder Symbol (außer dem Unterstrich) folgt, in einen Großbuchstaben und alle anderen Zeichen in Kleinbuchstaben geändert werden
STRING_SEG	Eine Spalte für jede Eingabespalte	Gibt das Zeichenfolgesegment zwischen zwei angegebenen Indizes zurück

<b>Makroname</b>	<b>Rückgabe</b>	<b>Beschreibung</b>
STRING_TAIL	Eine Spalte für jede Eingabespalte	Gibt die letzten <i>n</i> Zeichen jeder Zeichenfolge im angegebenen Datenbereich zurück
SUBSTR or SUBSTRING	Eine Spalte für jede Eingabespalte	Gibt Zeichen aus einer Zeichenfolge ab einer Anfangsposition zurück
UPPER	Eine Spalte für jede Eingabespalte	Konvertiert Zeichenfolgewerte in Großschreibung

## Datums- und Zeitfunktionen

<b>Makroname</b>	<b>Rückgabe</b>	<b>Beschreibung</b>
CURRENT_DATE	Eine Spalte für jede Eingabespalte	Gibt das aktuelle Datum in einem bestimmten format zurück
CURRENT_DAY	Eine Spalte für jede Eingabespalte	Gibt den aktuellen Tag des Monats als Zahl zwischen 1 und 31 zurück
CURRENT_JULIAN	Eine Spalte für jede Eingabespalte	Gibt die julianische Zahl für das aktuelle Datum zurück
CURRENT_MONTH	Eine Spalte für jede Eingabespalte	Gibt den aktuellen Monat des Jahres als Zahl zwischen 1 und 12 zurück
CURRENT_TIME	Eine Spalte für jede Eingabespalte	Gibt die aktuelle Uhrzeit als Zeichenfolge zurück
CURRENT_WEEKDAY	Eine Spalte für jede Eingabespalte	Gibt den aktuellen Wochentag als Zahl zwischen 0 und 6 zurück
CURRENT_YEAR	Eine Spalte für jede Eingabespalte	Gibt das aktuelle Jahr als Zahl zurück
DATE	Eine Spalte für jede Eingabespalte	Konvertiert eine Datumszeichenfolge in ein julianisches Datum
DATE_FORMAT	Eine Spalte für jede Eingabespalte	Wandelt Datumsformate um
DATE_JULIAN	Eine Spalte für jede Eingabespalte	Gibt das julianische Datum zurück
DATE_STRING	Eine Spalte für jede Eingabespalte	Gibt die Datumszeichenfolge des julianischen Datums zurück
DAY_BETWEEN	Eine Spalte für jede Eingabespalte	Gibt die Anzahl der Tage zwischen zwei Terminen zurück
DAY_FROMNOW	Eine Spalte für jede Eingabespalte	Gibt die Anzahl der Tage vom aktuellen Datum bis zum angegebenen Datum zurück

<b>Makroname</b>	<b>Rückgabe</b>	<b>Beschreibung</b>
DAY_INTERVAL	Eine Spalte für jede Eingabespalte	Gibt die Anzahl der Tage zwischen zwei Terminen zurück
DAYOF	Eine Spalte für jede Eingabespalte	Gibt den Tag des Monats als Zahl zurück
MONTHOF	Eine Spalte für jede Eingabespalte	Gibt den Monat des Jahres als Zahl zurück
WEEKDAY	Eine Spalte für jede Eingabespalte	Konvertiert ASCII-Datumszeichenfolgen in Wochentage
WEEKDAYOF	Eine Spalte für jede Eingabespalte	Gibt den Wochentag der Woche als Zahl zurück
YEAROF	Eine Spalte für jede Eingabespalte	Gibt das Jahr als Zahl zurück

## Gruppierungsfunktionen

<b>Makroname</b>	<b>Rückgabe</b>	<b>Beschreibung</b>
GROUPBY	Eine neue Spalte mit einem Wert für jede Zeile	Fasst mehrere Zeilen von Daten in einer Gruppe zusammen
GROUPBY_WHERE	Eine neue Spalte mit einem Wert für jede Zeile	Fasst mehrere Zeilen von Daten zusammen, die eine bestimmte Bedingung erfüllen und sich in einer Gruppe befinden

## Verschiedene Funktionen

<b>Makroname</b>	<b>Rückgabe</b>	<b>Beschreibung</b>
IF	Eine Spalte mit einem Wert für jede Zeile der kürzesten Eingabespalte	Leitet eine If-then-else-Anweisung einer bedingten Anweisung ein
ISERROR	Eine Spalte mit einem Wert für jede Zeile der kürzesten Eingabespalte	Gibt eine Eins zurück, wenn ein Wert in der Eingabezeile eine fehlerhafte Zelle (???) enthält, andernfalls eine Null
ISEVEN	Eine Spalte für jede Eingabespalte	Prüft, ob die Eingabewerte gerade (d. h. durch zwei teilbar) sind
ISODD	Eine Spalte für jede Eingabespalte	Prüft, ob die Eingabewerte ungerade (d. h. nicht durch zwei teilbar) sind
ROWNUM	Eine Spalte für jede Eingabespalte	Generiert fortlaufende Zahlen von eins bis zur Anzahl der Einträge

---

## Makrofunktionsparameter für IBM Campaign

Dieser Abschnitt beschreibt die Parameter und ihre Verwendung für Makrofunktionen in IBM Campaign.

### Formatspezifikationen

Dieser Abschnitt beschreibt das Format für einige häufig verwendete Parameter. Es gilt für alle Referenzen auf diese Parameter durch Makrofunktionsspezifikationen in diesem Kapitel.

#### Daten

Der Parameter data stellt eine Datenspalte dar, auf die eine Makrofunktion angewendet werden soll.

Es kann sich um eine Konstante oder um ein Feld handeln. Nähere Einzelheiten finden Sie im Abschnitt zur jeweiligen Makrofunktion.

**Anmerkung:** IBM Campaign unterstützt keine Berechnungen in mehreren Feldern gleichzeitig oder in einer Untergruppe von Zeilen.

Einige weitere Parameternamen verwenden dasselbe Format wie data. Die Beschreibungen dieser Parameter verweisen auf dieses Kapitel und Format.

#### Schlüsselwort

Der Parameter keyword steuert das Verhalten der Makrofunktion. Er zeigt an, dass ein Schlüsselwort angegeben werden kann (wenn es weggelassen wird, wird der Standardwert verwendet). Die Schlüsselwortoptionen werden für die jeweilige Makrofunktion in der folgenden Form aufgeführt:

**{choice1 | choice2 | choice3}**

Wählen Sie die Schlüsselwortoption aus, die zu dem gewünschten Verhalten führt. Die Standardoption wird in Fettschrift angezeigt. Es können zum Beispiel die folgenden Optionen angegeben sein:

**{RADIANs | DEGREEs}**

In diesem Fall sind die beiden folgenden Makrofunktionen gültig:

`COS(V1, RADIANs) COS(V1, DEGREEs)`

**Anmerkung:** Viele Makrofunktionen nehmen die Schlüsselwortparameter **{ALL | COL | ROW}** an. Diese Schlüsselwörter gelten nicht für IBM Campaign, da es sich bei den Eingabedaten immer um eine Einzelspalte oder ein Einzelfeld handelt. Das Makro verhält sich immer so, als ob das Schlüsselwort COL angegeben würde. Deshalb brauchen Sie diese Schlüsselwörter nicht anzugeben, wenn Sie IBM Campaign verwenden.

### Verwendung von Konstanten

Die meisten Makrofunktionsparameter verarbeiten numerische Konstanten oder Ausdrücke, deren Auswertung eine numerische Konstante ergibt (Makrofunktionen für Zeichenfolgen können Zeichenfolgekonstanten verarbeiten).

In Makrofunktionen, die alle Einträge einzeln durchgehen (etwa bei der Addition zweier numerischer Spalten), ist die Verwendung einer Konstante äquivalent zur

Angabe einer Spalte, die in jeder Zeile den betreffenden konstanten Wert enthält. Im Wesentlichen wird eine Konstante, die als ein Eingabeparameter angegeben ist, bis zur Länge der Eingabe expandiert.

Einige Makrofunktionen können ASCII-Zeichenfolgen und numerische Konstanten verarbeiten. Bei Parametern, die sowohl numerische Konstanten als auch ASCII-Zeichenfolgen verarbeiten können, ist dies im Abschnitt "Parameter" der jeweiligen Makrofunktion vermerkt.

Die folgende Tabelle enthält einige Beispiele.

Funktionsdefinition	Interpretation der Konstante
PERCENT_UTILIZ = (CURR_BAL*100)/ CREDIT_LIM	Die Konstante 100 wird als Spalte interpretiert, die dieselbe Zeilenzahl wie die Spalte CURR_BAL enthält, wobei jede Zeile die Konstante 100 enthält. Das abgeleitete Feld PERCENT_UTILIZ enthält jeden Wert von CURR_BAL, multipliziert mit 100 und dividiert durch jeden Wert von CREDIT_LIM.
NAME = STRING_CONCAT ("Mr. ", LAST_NAME)	Die Konstante "Mr." wird als Spalte interpretiert, die dieselbe Zeilenzahl wie die Spalte LAST_NAME enthält, wobei jede Zeile die Konstante "Mr." enthält. Im abgeleiteten Feld NAME ist jeder der Zeichenfolgen in LAST_NAME die Zeichenfolge "Mr." vorangestellt.

---

## Kapitel 2. Verwendung von Makros in IBM Interact

Dieses Kapitel enthält Informationen zur Verwendung von Makros in IBM Interact. Lesen Sie dieses Kapitel unbedingt, bevor Sie versuchen, Makros in IBM Interact zu verwenden.

### Formelhilfe und Syntaxprüfung von Makroausdrücken

**Wichtig:** Das Dialogfeld der **Formelhilfe** sowie die darin enthaltene Funktion zur Syntaxprüfung prüfen Makroausdrücke derzeit gemäß den von IBM Campaign unterstützten Makros. IBM Interact unterstützt jedoch nur eine Teilmenge der Makrofunktionalität von Campaign. Daher müssen Sie sicherstellen, dass die Makros und Schlüsselwörter (z. B. Schlüsselwörter für das Datumsformat) unterstützt werden, die in Interact verwendet werden. Suchen Sie im Kapitel zur Makroreferenz in diesem Handbuch nach Hinweisen in Bezug auf Interact.

---

### Zusammenfassung der Makrofunktionen für IBM Interact

Die Tabellen in den folgenden Abschnitten enthalten ausführliche Beschreibungen der Makros, die für IBM Interact spezifisch sind.

Ausführliche Referenzseiten zu den einzelnen Makrofunktionen in alphabetischer Reihenfolge werden in Kapitel 3, „IBM Makroreferenz“, auf Seite 17 bereitgestellt.

**Wichtig:** Makros können für IBM Campaign und IBM Interact oder nur für eines der beiden Produkte Anwendung finden. In den Makrobeschreibungen werden die Produkte angegeben, in denen das jeweilige Makro verfügbar ist.

„Makrofunktionsparameter für IBM Interact“ auf Seite 15 enthält Informationen zu den Eingabeparametern der Makrofunktionen für Interact.

### Statistische Funktionen

Makroname	Rückgabe	Beschreibung
AVG	Bei dem Schlüsselwort ALL ein Einzelwert in einer neuen Spalte; bei dem Schlüsselwort COL eine Spalte mit einem Einzelwert für jede Eingabespalte; bei dem Schlüsselwort ROW eine Spalte mit einem Wert für jede Zeile	Berechnet das arithmetische Mittel oder den Durchschnitt eines Zellenbereichs
MAX	Bei dem Schlüsselwort ALL ein Einzelwert in einer neuen Spalte; bei dem Schlüsselwort COL eine Spalte mit einem Einzelwert für jede Eingabespalte; bei dem Schlüsselwort ROW eine Spalte mit einem Wert für jede Zeile	Berechnet das Maximum eines Zellenbereichs

<b>Makroname</b>	<b>Rückgabe</b>	<b>Beschreibung</b>
MEAN	Bei dem Schlüsselwort ALL ein Einzelwert in einer neuen Spalte; bei dem Schlüsselwort COL eine Spalte mit einem Einzelwert für jede Eingabespalte; bei dem Schlüsselwort ROW eine Spalte mit einem Wert für jede Zeile	Berechnet das arithmetische Mittel oder den Durchschnitt eines Zellenbereichs
MIN	Bei dem Schlüsselwort ALL ein Einzelwert in einer neuen Spalte; bei dem Schlüsselwort COL eine Spalte mit einem Einzelwert für jede Eingabespalte; bei dem Schlüsselwort ROW eine Spalte mit einem Wert für jede Zeile	Berechnet das Minimum eines Zellenbereichs
STDV oder STDEV	Bei dem Schlüsselwort ALL ein Einzelwert in einer neuen Spalte; bei dem Schlüsselwort COL eine Spalte mit einem Einzelwert für jede Eingabespalte; bei dem Schlüsselwort ROW eine Spalte mit einem Wert für jede Zeile	Berechnet die Standardabweichung eines Zellenbereichs

## Mathematische und trigonometrische Funktionen

<b>Makroname</b>	<b>Rückgabe</b>	<b>Beschreibung</b>
AVG	Eine Spalte für jede Eingabespalte	Berechnet das arithmetische Mittel oder den Durchschnitt der Zellen im angegebenen Datenbereich.
MAX	Bei dem Schlüsselwort ALL ein Einzelwert in einer neuen Spalte; bei dem Schlüsselwort COL eine Spalte mit einem Einzelwert für jede Eingabespalte; bei dem Schlüsselwort ROW eine Spalte mit einem Wert für jede Zeile	Berechnet das Maximum eines Zellenbereichs
MEAN	Bei dem Schlüsselwort ALL ein Einzelwert in einer neuen Spalte; bei dem Schlüsselwort COL eine Spalte mit einem Einzelwert für jede Eingabespalte; bei dem Schlüsselwort ROW eine Spalte mit einem Wert für jede Zeile	Berechnet das arithmetische Mittel oder den Durchschnitt eines Zellenbereichs

<b>Makroname</b>	<b>Rückgabe</b>	<b>Beschreibung</b>
MIN	Bei dem Schlüsselwort ALL ein Einzelwert in einer neuen Spalte; bei dem Schlüsselwort COL eine Spalte mit einem Einzelwert für jede Eingabespalte; bei dem Schlüsselwort ROW eine Spalte mit einem Wert für jede Zeile	Berechnet das Minimum eines Zellenbereichs
STDV or STDEV	Bei dem Schlüsselwort ALL ein Einzelwert in einer neuen Spalte; bei dem Schlüsselwort COL eine Spalte mit einem Einzelwert für jede Eingabespalte; bei dem Schlüsselwort ROW eine Spalte mit einem Wert für jede Zeile	Berechnet die Standardabweichung eines Zellenbereichs
SUM	Bei dem Schlüsselwort ALL ein Einzelwert in einer neuen Spalte; bei dem Schlüsselwort COL eine Spalte mit einem Einzelwert für jede Eingabespalte; bei dem Schlüsselwort ROW eine Spalte mit einem Wert für jede Zeile	Berechnet die Summe eines Zellenbereichs.
TOTAL	Bei dem Schlüsselwort ALL ein Einzelwert in einer neuen Spalte; bei dem Schlüsselwort COL eine Spalte mit einem Einzelwert für jede Eingabespalte; bei dem Schlüsselwort ROW eine Spalte mit einem Wert für jede Zeile	Berechnet die Summe eines Zellenbereichs.

## Zeichenfolgenfunktionen

<b>Makroname</b>	<b>Rückgabe</b>	<b>Beschreibung</b>
LIKE	Eine Spalte für jede Eingabespalte	Ermittelt, ob eine Zeichenfolge mit einem angegebenen Muster übereinstimmt
LOWER	Eine Spalte für jede Eingabespalte	Konvertiert Zeichenfolgewerte in Kleinschreibung
LTRIM	Eine Spalte für jede Eingabespalte	Entfernt führende Leerzeichen aus jedem Zeichenfolgewart
NUMBER	Eine Spalte für jede Eingabespalte	Konvertiert ASCII-Zeichenfolgen für Uhrzeit- und Datumsangaben in numerische Werte
RTRIM	Eine Spalte für jede Eingabespalte	Entfernt nachfolgende Leerzeichen aus jedem Zeichenfolgewart

Makroname	Rückgabe	Beschreibung
STRING_CONCAT	Eine Spalte mit einem Wert für jede Zeile der kürzesten Eingabespalte	Verkettet Zeichenfolgen aus den angegebenen Datenbereichen
SUBSTR or SUBSTRING	Eine Spalte für jede Eingabespalte	Gibt Zeichen aus einer Zeichenfolge ab einer Anfangsposition zurück
UPPER	Eine Spalte für jede Eingabespalte	Konvertiert Zeichenfolgewerte in Großschreibung

## Datums- und Zeitfunktionen

Makroname	Rückgabe	Beschreibung
CURRENT_DATE	Eine Spalte für jede Eingabespalte	Gibt das aktuelle Datum in einem bestimmten format zurück
CURRENT_DAY	Eine Spalte für jede Eingabespalte	Gibt den aktuellen Tag des Monats als Zahl zwischen 1 und 31 zurück
CURRENT_MONTH	Eine Spalte für jede Eingabespalte	Gibt den aktuellen Monat des Jahres als Zahl zwischen 1 und 12 zurück
CURRENT_WEEKDAY	Eine Spalte für jede Eingabespalte	Gibt den aktuellen Wochentag des Monats als Zahl zwischen 0 und 6 zurück
CURRENT_YEAR	Eine Spalte für jede Eingabespalte	Gibt das aktuelle Jahr als Zahl zurück
DATE	Eine Spalte für jede Eingabespalte	Konvertiert eine Datumszeichenfolge in ein julianisches Datum
DATE_FORMAT	Eine Spalte für jede Eingabespalte	Wandelt Datumsformate um

## Verschiedene Funktionen

Makroname	Rückgabe	Beschreibung
EXTERNALCALLOUT	Werte, die von der über die ExternalCallout-API erstellten kundenspezifischen Anwendung definiert worden sind	Ruft eine kundenspezifische Anwendung auf, die mithilfe der ExternalCallout-API geschrieben wurde.  Weitere Informationen finden Sie im <i>IBM Interact-Administratorhandbuch</i> .
IF	Eine Spalte mit einem Wert für jede Zeile der kürzesten Eingabespalte	Leitet eine If-then-else-Anweisung einer bedingten Anweisung ein

---

## Makrofunktionsparameter für IBM Interact

Dieser Abschnitt beschreibt die Parameter und ihre Verwendung für Makrofunktionen in IBM Interact.

### Formatspezifikationen

Dieser Abschnitt beschreibt das Format für einige häufig verwendete Parameter. Es gilt für alle Referenzen auf diese Parameter durch Makrofunktionsspezifikationen in diesem Abschnitt.

#### Daten

Der Parameter data stellt eine Datenspalte dar, auf die eine Makrofunktion angewendet werden soll. Es kann sich um eine Konstante oder um ein Feld handeln. Nähere Einzelheiten finden Sie im Abschnitt zur jeweiligen Makrofunktion.

**Anmerkung:** IBM Interact unterstützt keine Berechnungen in mehreren Feldern gleichzeitig oder in einer Untergruppe von Zeilen.

Einige weitere Parameternamen verwenden dasselbe Format wie data. Die Beschreibungen dieser Parameter verweisen auf dieses Kapitel und Format.

#### Schlüsselwort

Der Parameter keyword steuert das Verhalten der Makrofunktion. Er zeigt an, dass ein Schlüsselwort angegeben werden kann (wenn es weggelassen wird, wird der Standardwert verwendet). Die Schlüsselwortoptionen werden für die jeweilige Makrofunktion in der folgenden Form aufgeführt:

```
{choice1 | choice2 | choice3}
```

Wählen Sie die Schlüsselwortoption aus, die zu dem gewünschten Verhalten führt. Die Standardoption wird in Fettschrift angezeigt. Es können zum Beispiel die folgenden Optionen angegeben sein:

```
{RADIANS | DEGREES}
```

In diesem Fall sind die beiden folgenden Makrofunktionen gültig:

```
COS(V1, RADIANS) COS(V1, DEGREES)
```

**Anmerkung:** Viele Makrofunktionen setzen die Schlüsselwortparameter {ALL | COL | ROW} ein. Diese Schlüsselwörter gelten nicht für IBM Interact, da es sich bei den Eingabedaten immer um eine Einzelspalte oder ein Einzelfeld handelt. Das Makro verhält sich immer so, als ob das Schlüsselwort COL angegeben würde. Deshalb brauchen Sie diese Schlüsselwörter nicht anzugeben, wenn Sie IBM Interact verwenden.

### Verwendung von Konstanten

Die meisten Makrofunktionsparameter verarbeiten numerische Konstanten oder Ausdrücke, deren Auswertung eine numerische Konstante ergibt (Makrofunktionen für Zeichenfolgen können Zeichenfolgekonstanten verarbeiten).

In Makrofunktionen, die alle Einträge einzeln durchgehen (etwa bei der Addition zweier numerischer Spalten), ist die Verwendung einer Konstante äquivalent zur

Angabe einer Spalte, die in jeder Zeile den betreffenden konstanten Wert enthält. Im Wesentlichen wird eine Konstante, die als ein Eingabeparameter angegeben ist, bis zur Länge der Eingabe expandiert.

Einige Makrofunktionen können ASCII-Zeichenfolgen und numerische Konstanten verarbeiten. Bei Parametern, die sowohl numerische Konstanten als auch ASCII-Zeichenfolgen verarbeiten können, ist dies im Abschnitt "Parameter" der jeweiligen Makrofunktion vermerkt.

Die folgende Tabelle enthält einige Beispiele.

Funktionsdefinition	Interpretation der Konstante
PERCENT_UTILIZ = (CURR_BAL*100)/ CREDIT_LIM	Die Konstante 100 wird als Spalte interpretiert, die dieselbe Zeilenzahl wie die Spalte CURR_BAL enthält, wobei jede Zeile die Konstante 100 enthält. Das abgeleitete Feld PERCENT_UTILIZ enthält jeden Wert von CURR_BAL, multipliziert mit 100 und dividiert durch jeden Wert von CREDIT_LIM.
NAME = STRING_CONCAT ("Mr.", LAST_NAME)	Die Konstante "Mr." wird als Spalte interpretiert, die dieselbe Zeilenzahl wie die Spalte LAST_NAME enthält, wobei jede Zeile die Konstante "Mr." enthält. Im abgeleiteten Feld NAME ist jeder der Zeichenfolgen in LAST_NAME die Zeichenfolge "Mr." vorangestellt.

**Anmerkung:** Konstanten wie DT\_DELIM\_M\_D\_Y erfordern einfache Anführungszeichen.

---

## Kapitel 3. IBM Makroreferenz

In diesem Abschnitt werden die einzelnen Makros beschrieben, die zur Verwendung in IBM Campaign und/oder IBM Interact verfügbar sind. Makros werden in alphabetischer Reihenfolge aufgelistet.

**Wichtig:** Verwenden Sie keine Funktionsnamen oder Schlüsselwörter aus der IBM Makrosprache für Spaltenüberschriften von Benutzertabellen in IBM Campaign, unabhängig davon, ob deren Zuordnung auf einer Datenbank oder einer unstrukturierten Datei beruht. Diese reservierten Wörter können Fehler verursachen, wenn sie in Spaltenüberschriften zugeordneter Tabellen verwendet werden.

---

### Gültige Datumsformatschlüsselwörter

In der folgenden Tabelle sehen Sie die Schlüsselwörter für gültige Formate, einschließlich einer Beschreibung und eines Beispiels.

Keyword	Beschreibung	Beispiel(e)
MM	2-stelliger Monat	01, 02, 03, ..., 12
MMDD	2-stelliger Monat und 2-stelliger Tag	0331 steht für 31. März
MMDDYY	2-stelliger Monat, 2-stelliger Tag und 2-stelliges Jahr	033170 steht für 31. März 1970
MMDDYYYY	2-stelliger Monat, 2-stelliger Tag und 4-stelliges Jahr	03311970 steht für 31. März 1970"
DELIM_M_D	Monate mit Begrenzer, gefolgt von der Tagesangabe	March 31, 3/31 oder 03-31
DELIM_M_D_Y	Jeder Monat, Tag oder jedes Jahr mit Begrenzer	March 31, 1970 oder 3/31/70
DELIM_M_D_YYYY	Belieb. Monat, Tag und 4-stelliges Jahr, getrennt durch Begrenzer	März 31 1970 oder 3/31/1970
DELIM_Y_M	Jedes Jahr mit Begrenzer, gefolgt von einem Monat	70 März, 70-3 oder 1970/3
DELIM_Y_M_D	Jedes Jahr und jeder Monat und Tag mit Begrenzer	1970 Mar 31 oder 70/3/31
YYMMM	2-stelliges Jahr und Monat mit 3 Buchstaben	70MAR
YYMMDD	2-stelliges Jahr, Monat mit 3 Buchstaben und zweistelliger Tag	70MAR31
YY	2-stelliges Jahr	70
YYMM	2-stelliges Jahr und 2-stelliger Monat	7003
YYMMDD	2-stelliges Jahr, 2-stelliger Monat und 2-stelliger Tag	700331
YYYYMMM	4-stelliges Jahr und Monat mit 3 Buchstaben	1970MAR

<b>Keyword</b>	<b>Beschreibung</b>	<b>Beispiel(e)</b>
YYYYMMDD	4-stelliges Jahr, Monat mit 3 Buchstaben und 2-stelliger Tag	1970MAR31
YYYY	4-stelliges Jahr	1970
YYYYMM	4-stelliges Jahr und 2-stelliger Monat	197003
YYYYMMDD	4-stelliges Jahr, 2-stelliger Monat und 2-stelliger Tag	19700331
DELIM_M_Y	Jeder Monat mit Begrenzer, gefolgt von einem Jahr	3-70, 3/70, Mar 70, March 1970
DELIM_D_M	Jeder Tag mit Begrenzer, gefolgt von einem Monat	31-3, 31/3, 31 March
DELIM_D_M_Y	Jeder Tag, Monat und jedes Jahr mit Begrenzer	31-MAR-70, 31/3/1970, 31 03 70
DD	2-stelliger Tag	31
DDMMM	2-stelliger Tag und Monat mit 3 Buchstaben	31MAR
DDMMYY	2-stelliger Tag, Monat mit 3 Buchstaben und 2-stelliges Jahr	31MAR70
DDMMYYYY	2-stelliger Tag, Monat mit 3 Buchstaben und 4-stelliges Jahr	31MAR1970
DDMM	2-stelliger Tag und 2-stelliger Monat	3103
DDMMYY	2-stelliger Tag, 2-stelliger Monat und 2-stelliges Jahr	310370
DDMMYYYY	2-stelliger Tag, 2-stelliger Monat und 4-stelliges Jahr	31031970
MMYY	2-stelliger Monat und 2-stelliges Jahr	0370
MMYYYY	2-stelliger Monat und 4-stelliges Jahr	031970
MMM	Monat mit 3 Buchstaben	MAR
MMMDD	Monat mit drei Buchstaben und 2-stelliger Tag	MAR31
MMDDYY	Monat mit drei Buchstaben, 2-stelliger Tag und 2-stelliges Jahr	MAR3170
MMDDYYYY	Monat mit drei Buchstaben, 2-stelliger Tag und 4-stelliges Jahr	MAR311970
MMYY	Monat mit drei Buchstaben und 2-stelliges Jahr	MAR70
MMYYYY	Monat mit drei Buchstaben und 4-stelliges Jahr	MAR1970
MONTH	Monat des Jahres	Januar, Februar, März usw. oder Jan, Feb, März usw.

Keyword	Beschreibung	Beispiel(e)
WEEKDAY	Wochentag	Sonntag, Montag, Dienstag usw. (Sonntag = 0)
WKD	Abgekürzter Wochentag	So, Mo, Di usw. (So = 0)

## ABS-Makro

Das ABS-Makro ist nur in IBM Campaign verfügbar.

### Syntax

ABS(data)

### Parameter

data

Die numerischen Werte, deren absoluter Wert berechnet werden soll. Dieser Parameter kann ein konstanter Wert, eine Spalte, ein Zellenbereich oder ein Ausdruck sein, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

ABS berechnet den absoluten Wert der Zahlen im angegebenen Datenbereich. Der absolute Wert einer Zahl ist ihr Wert ohne das Vorzeichen (d. h., positive Zahlen werden nicht geändert, negative Zahlen werden als positive Zahlen zurückgegeben). ABS gibt für jede Eingabespalte eine neue Spalte zurück, die den absoluten Wert der Zahlen in der entsprechenden Eingabespalte enthält.

### Beispiele

TEMP = ABS(-3) oder TEMP = ABS(3)
Erstellt eine Spalte TEMP, die den Wert 3 enthält.
TEMP = ABS(V1)
Erstellt eine Spalte TEMP, in der jeder Wert den absoluten Wert des Inhalts von Spalte V1 darstellt.
TEMP = ABS(V1:V3)
Erstellt drei neue Spalten TEMP, VX und VY. Die Werte in der Spalte TEMP sind die absoluten Werte des Inhalts von Spalte V1, die Werte von Spalte VX sind die absoluten Werte des Inhalts von Spalte V2 und die Werte von Spalte VY sind die absoluten Werte des Inhalts von Spalte V3.
TEMP = ABS(V1[10:20])
Erstellt eine Spalte TEMP, in der die ersten 11 Zellen die absoluten Werte der Werte in Zeile 10-20 von Spalte V1 darstellen. Die anderen Zellen in TEMP sind leer.

TEMP = ABS(V1[1:5]:V2)

Erstellt zwei neue Spalten TEMP und VX, die jeweils die Werte in Zeile 1-5 enthalten (die anderen Zellen sind leer). Die Werte in der Spalte TEMP sind die absoluten Werte der entsprechenden Zeilen von Spalte V1, die Werte in der Spalte VX sind die absoluten Werte der entsprechenden Zeilen von Spalte V2.

## Zugehörige Funktionen

Funktion	Beschreibung
SIGN	Berechnet das Vorzeichen (positiv oder negativ) der Werte im angegebenen Datenbereich.

---

## ACOS-Makro

Das ACOS-Makro Makro ist nur in IBM Campaign verfügbar.

### Syntax

ACOS( data [, units\_keyword ])

### Parameter

Daten

Die numerischen Werte, deren Arkuskosinus berechnet werden soll. Dieser Parameter kann ein konstanter Wert, eine Spalte, ein Zellenbereich oder ein Ausdruck sein, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

units\_keyword

Dieses optionale Schlüsselwort legt fest, ob die Eingabewerte und Ergebnisse als Grad oder als Radiant interpretiert werden. Wählen Sie einen der folgenden Werte aus:

RADIAN - Führt die Berechnungen in Radiant aus (Standard)

DEGREE - Führt die Berechnungen in Grad aus

Wird dieser Parameter nicht angegeben, ist der Standard Radiant. (Die Umrechnung von Radiant in Grad erfolgt durch Division durch PI und Multiplikation mit 180.)

Weitere Informationen zur Verwendung von Schlüsselwörtern in IBM Campaign finden Sie in „Formatspezifikationen“ auf Seite 9.

### Beschreibung

ACOS berechnet den Arkuskosinus der Werte im angegebenen Datenbereich. Der Arkuskosinus ist der Winkel, dessen Kosinus der Inhalt der jeweiligen Zelle ist. ACOS gibt für jede Eingabespalte eine neue Spalte zurück, die den Arkuskosinus der Zahlen in der entsprechenden Eingabespalte enthält.

Wenn das Schlüsselwort RADIAN verwendet wird, gibt ACOS Werte im Bereich 0 bis  $\pi$  zurück. Wenn das Schlüsselwort DEGREE verwendet wird, gibt ACOS Werte im Bereich 0 bis 180 zurück.

**Anmerkung:** Der Zelleninhalt jeder angegebenen Spalte muss Werte zwischen -1,0 und 1,0 (einschließlich) aufweisen. Andernfalls wird für jede ungültige Eingabe eine leere Zelle zurückgegeben.

## Beispiele

TEMP = ACOS(0) oder TEMP = ACOS(0, 0) oder TEMP = ACOS(0, RADIAN)
Erstellt eine Spalte TEMP, die den Wert 1.571 ( $\pi/2$ Radiant) enthält.
TEMP = ACOS(0, 1) oder TEMP = ACOS(0, DEGREE)
Erstellt eine Spalte TEMP, die den Wert 90 (Grad) enthält.
TEMP = ACOS(V1)
Erstellt eine Spalte TEMP, in der jeder Wert den Arkuskosinus (in Radiant) des Inhalts von Spalte V1 darstellt.
TEMP = ACOS(V1:V3, 1)
Erstellt drei neue Spalten TEMP, VX und VY. Die Werte in der Spalte TEMP sind jeweils der Arkuskosinus des Inhalts von Spalte V1, die Werte von Spalte VX sind der Arkuskosinus des Inhalts von Spalte V2 und die Werte von Spalte VY sind der Arkuskosinus des Inhalts von Spalte V3. Alle Werte sind in Grad angegeben.
TEMP = ACOS(V1[10:20])
Erstellt eine neue Spalte TEMP, in der die ersten 11 Zellen jeweils den Arkuskosinus der Werte in Zeile 10-20 von Spalte V1 darstellen. Die anderen Zellen in TEMP sind leer.
TEMP = ACOS(V1[1:5]:V2)
Erstellt zwei neue Spalten TEMP und VX, die jeweils die Werte in Zeile 1-5 enthalten (die anderen Zellen sind leer). Die Werte in der Spalte TEMP sind der Arkuskosinus der entsprechenden Zeilen von Spalte V1, die Werte in der Spalte VX sind der Arkuskosinus der entsprechenden Zeilen von Spalte V2. Alle Werte sind in Radiant angegeben.

## Zugehörige Funktionen

Funktion	Beschreibung
ACOT	Berechnet den Arkuskotangens des Inhalts des angegebenen Datenbereichs.
ASIN	Berechnet den Arkussinus des Inhalts des angegebenen Datenbereichs.
ATAN	Berechnet den Arkustangens des Inhalts des angegebenen Datenbereichs.
COS	Berechnet den Kosinus des Inhalts des angegebenen Datenbereichs.

## ACOT-Makro

Das ACOT-Makro Makro ist nur in IBM Campaign verfügbar.

## Syntax

ACOT(data [, units\_keyword])

## Parameter

data

Die numerischen Werte, deren Arkuskotangens berechnet werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

units\_keyword

Dieses optionale Schlüsselwort legt fest, ob die Eingabewerte und Ergebnisse als Grad oder als Radiant interpretiert werden. Wählen Sie eine der folgenden Angaben aus:

RADIAN - Führt die Berechnungen in Radiant aus (Standard)

DEGREE - Führt die Berechnungen in Grad aus

Wird dieser Parameter nicht angegeben, ist der Standard Radiant. (Die Umrechnung von Radiant in Grad erfolgt durch Division durch  $\pi$  und Multiplikation mit 180.)

Weitere Informationen zur Verwendung von Schlüsselwörtern in IBM Campaign finden Sie in „Formatspezifikationen“ auf Seite 9.

## Beschreibung

ACOT gibt den Winkel zurück, dessen Kotangens der Inhalt der jeweiligen Zelle ist. ACOT gibt für jede Eingabespalte eine neue Spalte zurück, die den Arkuskotangens der Zahlen in der entsprechenden Eingabespalte enthält. Dabei werden 64-Bit-Gleitkommazahlen verwendet.

## Beispiele

TEMP = ACOT(0.5) oder TEMP = ACOT(0.5, 0) oder TEMP = ACOT(0.5, RADIAN) Erstellt eine Spalte TEMP, die den Wert 2.157 (Radiant) enthält.
TEMP = ACOT(1, 1) oder TEMP = ACOT(1, DEGREE) Erstellt eine Spalte TEMP, die den Wert 0.022 (1/45) Grad enthält.
TEMP = ACOT(0) Erstellt eine Spalte TEMP, die den Wert MAX32_Float in Radiant enthält.
TEMP = ACOT(V1) Erstellt eine neue Spalte TEMP, in der jeder Wert den Arkuskotangens (in Radiant) des Inhalts von Spalte V1 darstellt.

TEMP = ACOT(V1:V3, 1)
Erstellt drei neue Spalten TEMP, VX und VY. Die Werte in der Spalte TEMP sind jeweils der Arkuskotangens des Inhalts von Spalte V1, die Werte von Spalte VX sind der Arkuskotangens des Inhalts von Spalte V2 und die Werte von Spalte VY sind der Arkuskotangens des Inhalts von Spalte V3. Alle Werte sind in Grad angegeben.
TEMP = ACOT(V1[10:20])
Erstellt eine neue Spalte TEMP, in der die ersten 11 Zellen jeweils den Arkuskotangens der Werte in Zeile 10-20 von Spalte V1 darstellen. Die anderen Zellen in TEMP sind leer.
TEMP = ACOT(V1[1:5]:V2)
Erstellt zwei neue Spalten TEMP und VX, die jeweils die Werte in Zeile 1-5 enthalten (die anderen Zellen sind leer). Die Werte in der Spalte TEMP sind der Arkuskotangens der entsprechenden Zeilen von Spalte V1, die Werte in der Spalte VX sind der Arkuskotangens der entsprechenden Zeilen von Spalte V2. Alle Werte sind in Radiant angegeben.

## Zugehörige Funktionen

Funktion	Beschreibung
ACOS	Berechnet den Arkuskosinus des Inhalts des angegebenen Datenbereichs.
ASIN	Berechnet den Arkussinus des Inhalts des angegebenen Datenbereichs.
ATAN	Berechnet den Arkustangens des Inhalts des angegebenen Datenbereichs.
COT	Berechnet den Kotangens des Inhalts des angegebenen Datenbereichs.

---

## ADD\_MONTHS-Makro

Das ADD\_MONTHS-Makro ist in IBM Campaign verfügbar

### Syntax

ADD\_MONTHS(months, date\_string [, input\_format])

### Parameter

months

Eine Ganzzahl, die eine Anzahl von Monaten darstellt, die zu [Datumszeichenfolge] addiert werden soll.

date\_string

Eine Zeichenfolge, die ein gültiges Datum in dem Format "DELIM\_M\_D\_Y" oder in dem durch das optionale Argument "[input\_format]" angegebenen Format darstellt.

input\_format

Das Format, das für das berechnete Datum verwendet werden soll. Eine Liste der unterstützten Datumsformate finden Sie unter der Funktion DATE\_FORMAT. Das

Argument `input_format` legt nicht nur das Format der Eingabezeichenfolge, sondern auch das Format der Ausgabezeichenfolge fest.

## Beschreibung

`ADD_MONTHS` gibt ein Datum zurück, das sich aus der Addition der angegebenen Anzahl von Monaten zu der angegebenen Datumszeichenfolge ergibt. Das Datum wird in dem Standardformat "DELIM\_M\_D\_Y" oder in dem durch das optionale Argument [Eingabeformat] angegebenen Format zurückgegeben. Mithilfe von `DATE_FORMAT` können Sie ein anderes Ausgabeformat festlegen.

Wenn die Erhöhung des Monats um die angegebene Anzahl von Monaten ein ungültiges Datum erzeugt, wird als Ergebnis der letzte Tag des Monats berechnet, wie im letzten der folgenden Beispiele zu sehen. Gegebenenfalls werden auch Schaltjahre berücksichtigt. Wenn beispielsweise zum 31. Januar 2012 ein Monat addiert wird, ergibt sich der 29. Februar 2012.

## Beispiele

`ADD_MONTHS(12, '06-25-11')` addiert zu dem angegebenen Datum ein Jahr (12 Monate) und gibt das Datum 06-25-12 zurück.

`ADD_MONTHS(3, '2011-06-25', DT_DELIM_Y_M_D)` addiert zu dem angegebenen Datum drei Monate und gibt das Datum 2011-09-25 zurück.

`ADD_MONTHS(1, '02-28-2011')` gibt das Datum 03-28-2011 zurück.

`ADD_MONTHS(1, '03-31-2012')` gibt das Datum 04-30-2012 zurück.

## Zugehörige Funktionen

Funktion	Beschreibung
<code>DATE</code>	Wandelt eine Datumszeichenfolge in ein julianisches Datum um.
<code>DATE_FORMAT</code>	Wandelt ein Datum von <code>input_format</code> in <code>output_format</code> um.

---

## AND-Makro

Das AND-Makro ist in IBM Campaign und IBM Interact verfügbar.

### Syntax

```
data1 AND data2 data1 && data2
```

### Parameter

`data1`

Die Zahlen, die durch logisches Und mit den Werten in `data2` verknüpft werden sollen. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makro funktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

data2

Die Zahl(en), die durch logisches Und mit den Werten in data1 verknüpft werden soll(en). Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Die Anzahl der Spalten in data2 muss mit der Anzahl der Spalten in data1 übereinstimmen, es sei denn, bei data2 handelt es sich um eine Konstante. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

## Beschreibung

AND berechnet das logische Und zwischen den beiden angegebenen Datenbereichen. AND gibt für jede Eingabespalte eine neue Spalte zurück, wobei jeweils die entsprechende Spalte von data1 durch logisches Und mit der entsprechenden Spalte von data2 verknüpft wird (d. h., die erste Spalte von data1 wird durch logisches Und mit der ersten Spalte von data2 verknüpft, die zweite Spalte mit der zweiten Spalte usw.).

Wenn es sich bei data2 um eine Konstante handelt, wird jeder Wert in data1 durch logisches Und mit dem betreffenden Wert verknüpft. Wenn data2 eine oder mehrere Spalten enthält, werden die Berechnungen zeilenweise zwischen einer Spalte von data1 und einer Spalte von data2 ausgeführt. Die erste Zeile von data1 wird durch logisches Und mit der ersten Zeile von data2 verknüpft, die zweite Zeile mit der zweiten Zeile usw. Diese zeilenweise ausgeführte Berechnung erzeugt für jede Zeile ein Ergebnis, bis zum letzten Wert der kürzesten Spalte.

**Anmerkung:** Der Operator AND kann durch ein doppeltes Et-Zeichen (&&) abgekürzt werden. Mit dem doppelten Et-Zeichen können die beiden Argumente getrennt werden (statt z. B. V1 AND 3 anzugeben, können Sie einfach V1&&3 eingeben).

## Beispiele

TEMP = 1 AND 8 oder TEMP = 1 && 8

Erstellt eine neue Spalte TEMP, die den Wert eins enthält (jede Zahl ungleich null wird als eins behandelt).

TEMP = V1 && 1

Erstellt eine neue Spalte TEMP, die für jeden Wert von Spalte V1 den Wert eins enthält.

TEMP = V1 && V1

Erstellt eine neue Spalte TEMP, die für jeden Wert ungleich null in Spalte V1 den Wert eins und für jede Null in Spalte V1 den Wert null enthält.

TEMP = V1 && V2

Erstellt eine neue Spalte TEMP, in der jeder Zeilenwert von Spalte V1 durch logisches Und mit dem entsprechenden Zeilenwert von Spalte V2 verknüpft ist.

TEMP = V1:V3 && V4:V6

Erstellt drei neue Spalten TEMP, VX und VY. In der Spalte TEMP sind die Werte von Spalte V1 durch logisches Und mit den entsprechenden Zeilenwerten von Spalte V4 verknüpft. In der Spalte VX sind die Werte von Spalte V2 durch logisches Und mit den Werten von Spalte V5 verknüpft. In der Spalte VY sind die Werte von Spalte V3 durch logisches Und mit den Werten von Spalte V6 verknüpft.

TEMP = V1[10:20] && V2 oder TEMP = V1[10:20] && V2[1:11]

Erstellt eine neue Spalte TEMP, in deren ersten 11 Zellen die Werte in Zeile 10-20 von Spalte V1 durch logisches Und mit den Werten in Zeile 1-11 von Spalte V2 verknüpft sind. Die anderen Zellen in TEMP sind leer.

## Zugehörige Funktionen

Funktion	Beschreibung
NOT	Berechnet das logische Nicht des Inhalts des angegebenen Datenbereichs.
ODER	Berechnet das logische Oder zwischen den beiden angegebenen Datenbereichen.

---

## ASIN-Makro

Das ASIN-Makro Makro ist nur in IBM Campaign verfügbar.

### Syntax

ASIN(data [, units\_keyword])

### Parameter

data

Die numerischen Werte, deren Arkussinus berechnet werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

units\_keyword

Dieses optionale Schlüsselwort legt fest, ob die Eingabewerte und Ergebnisse als Grad oder als Radiant interpretiert werden. Wählen Sie eine der folgenden Angaben aus:

RADIAN - Führt die Berechnungen in Radiant aus (Standard)

DEGREE - Führt die Berechnungen in Grad aus

Wird dieser Parameter nicht angegeben, ist der Standard Radiant. (Die Umrechnung von Radiant in Grad erfolgt durch Division durch PI und Multiplikation mit 180.)

Weitere Informationen zur Verwendung von Schlüsselwörtern in IBM Campaign finden Sie in „Formatspezifikationen“ auf Seite 9.

### Beschreibung

ASIN berechnet den Arkussinus der Werte im angegebenen Datenbereich. Der Arkussinus ist der Winkel, dessen Sinus der Inhalt der jeweiligen Zelle ist. ASIN gibt

für jede Eingabespalte eine neue Spalte zurück, die den Arkussinus der Zahlen in der entsprechenden Eingabespalte enthält.

Wenn das Schlüsselwort RADIAN verwendet wird, gibt ASIN Werte im Bereich  $-\pi/2$  bis  $\pi/2$  zurück. Wenn das Schlüsselwort DEGREE verwendet wird, gibt ASIN Werte im Bereich -90 bis 90 zurück.

**Anmerkung:** Der Zelleninhalt jeder angegebenen Spalte muss Werte zwischen -1,0 und 1,0 (einschließlich) aufweisen. Andernfalls wird für jede ungültige Eingabe ??? zurückgegeben.

## Beispiele

TEMP = ASIN(0.5) oder TEMP = ASIN(0.5, 0) oder TEMP = ASIN(0.5, RADIAN)
Erstellt eine neue Spalte TEMP, die den Wert 0.524 ( $\pi/6$ Radiant) enthält.
TEMP = ASIN(0.5, 1) oder TEMP = ASIN(0.5, DEGREE)
Erstellt eine neue Spalte TEMP, die den Wert 30 (Grad) enthält.
TEMP = ASIN(V1)
Erstellt eine neue Spalte TEMP, in der jeder Wert den Arkussinus (in Radiant) des Inhalts von Spalte V1 darstellt.
TEMP = ASIN(V1:V3, 1)
Erstellt drei neue Spalten TEMP, VX und VY. Die Werte in der Spalte TEMP sind jeweils der Arkussinus des Inhalts von Spalte V1, die Werte von Spalte VX sind der Arkussinus des Inhalts von Spalte V2 und die Werte von Spalte VY sind der Arkussinus des Inhalts von Spalte V3. Alle Werte sind in Grad angegeben.
TEMP = ASIN(V1[10:20])
Erstellt eine neue Spalte TEMP, in der die ersten 11 Zellen jeweils den Arkussinus der Werte in Zeile 10-20 von Spalte V1 darstellen. Die anderen Zellen in TEMP sind leer.
TEMP = ASIN(V1[1:5]:V2)
Erstellt zwei neue Spalten TEMP und VX, die jeweils die Werte in Zeile 1-5 enthalten (die anderen Zellen sind leer). Die Werte in der Spalte TEMP sind der Arkussinus der entsprechenden Zeilen von Spalte V1, die Werte in der Spalte VX sind der Arkussinus der entsprechenden Zeilen von Spalte V2. Alle Werte sind in Radiant angegeben.

## Zugehörige Funktionen

Funktion	Beschreibung
ACOS	Berechnet den Arkuskosinus des Inhalts des angegebenen Datenbereichs.
ACOT	Berechnet den Arkuskotangens des Inhalts des angegebenen Datenbereichs.
ATAN	Berechnet den Arkustangens des Inhalts des angegebenen Datenbereichs.
SIN	Berechnet den Sinus des Inhalts des angegebenen Datenbereichs.

---

## ATAN-Makro

Das ATAN-Makro Makro ist nur in IBM Campaign verfügbar.

### Syntax

ATAN(data [, units\_keyword])

### Parameter

data

Die numerischen Werte, deren Arkustangens berechnet werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

units\_keyword

Dieses optionale Schlüsselwort legt fest, ob die Eingabewerte und Ergebnisse als Grad oder als Radiant interpretiert werden. Wählen Sie eine der folgenden Angaben aus:

RADIAN - Führt die Berechnungen in Radiant aus (Standard)

DEGREE - Führt die Berechnungen in Grad aus

Wird dieser Parameter nicht angegeben, ist der Standard Radiant. (Die Umrechnung von Radiant in Grad erfolgt durch Division durch  $\pi$  und Multiplikation mit 180.)

Weitere Informationen zur Verwendung von Schlüsselwörtern in IBM Campaign finden Sie in „Formatspezifikationen“ auf Seite 9.

### Beschreibung

ATAN berechnet den Arkustangens der Werte im angegebenen Datenbereich. Der Arkustangens ist der Winkel, dessen Tangens der Inhalt der jeweiligen Zelle ist. ATAN gibt für jede Eingabespalte eine neue Spalte zurück, die den Arkustangens der Zahlen in der entsprechenden Eingabespalte enthält.

Wenn das Schlüsselwort RADIAN verwendet wird, gibt ATAN Werte im Bereich  $-\pi/2$  bis  $\pi/2$  zurück. Wenn das Schlüsselwort DEGREE verwendet wird, gibt ATAN Werte im Bereich -90 bis 90 zurück.

### Beispiele

TEMP = ATAN(1) oder TEMP = ATAN(1, 0) oder TEMP = ATAN(1, RADIAN)
Erstellt eine neue Spalte TEMP, die den Wert 0.785 ( $\pi/4$ Radiant) enthält.
TEMP = ATAN(1, 1) oder TEMP = ATAN(1, DEGREE)
Erstellt eine neue Spalte TEMP, die den Wert 45 (Grad) enthält.

TEMP = ATAN(V1)
Erstellt eine neue Spalte TEMP, in der jeder Wert den Arkustangens (in Radiant) des Inhalts von Spalte V1 darstellt.
TEMP = ATAN(V1:V3, 1)
Erstellt drei neue Spalten TEMP, VX und VY. Die Werte in der Spalte TEMP sind jeweils der Arkustangens des Inhalts von Spalte V1, die Werte von Spalte VX sind der Arkustangens des Inhalts von Spalte V2 und die Werte von Spalte VY sind der Arkustangens des Inhalts von Spalte V3. Alle Werte sind in Grad angegeben.
TEMP = ATAN(V1[10:20])
Erstellt eine neue Spalte TEMP, in der die ersten 11 Zellen jeweils den Arkustangens der Werte in Zeile 10-20 von Spalte V1 darstellen. Die anderen Zellen in TEMP sind leer.
TEMP = ATAN(V1[1:5]:V2)
Erstellt zwei neue Spalten TEMP und VX, die jeweils die Werte in Zeile 1-5 enthalten (die anderen Zellen sind leer). Die Werte in der Spalte TEMP sind der Arkustangens der entsprechenden Zeilen von Spalte V1, die Werte in der Spalte VX sind der Arkustangens der entsprechenden Zeilen von Spalte V2. Alle Werte sind in Radiant angegeben.

## Zugehörige Funktionen

Funktion	Beschreibung
ACOS	Berechnet den Arkuskosinus des Inhalts des angegebenen Datenbereichs.
ASIN	Berechnet den Arkussinus des Inhalts des angegebenen Datenbereichs.
ATAN	Berechnet den Arkustangens des Inhalts des angegebenen Datenbereichs.
TAN	Berechnet den Tangens des Inhalts des angegebenen Datenbereichs.

---

## AVG-Makro

Das AVG-Makro Makro ist in IBM Campaign und IBM Interact verfügbar.

### Syntax

AVG(data [, keyword])

### Parameter

data

Die numerischen Werte, deren arithmetisches Mittel berechnet werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen der obigen Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

keyword

Dieses optionale Schlüsselwort legt fest, wie die Berechnung auf den Eingabedatenbereich angewendet wird. Wählen Sie eines der folgenden Schlüsselwörter aus:

ALL - Wendet die Berechnung auf alle Zellen in data an (Standard)

COL - Führt die Berechnung für jede Spalte von data gesondert aus

ROW - Führt die Berechnung für jede Zeile von data gesondert aus

Weitere Informationen zur Verwendung von Schlüsselwörtern in IBM Campaign finden Sie in „Formatspezifikationen“ auf Seite 9.

**Anmerkung:** Viele Makrofunktionen setzen die Schlüsselwortparameter {ALL | COL | ROW} ein. Diese Schlüsselwörter gelten nicht für **IBM Campaign**, da es sich bei den Eingabedaten immer um eine Einzelspalte oder ein Einzelfeld handelt. Das Makro verhält sich immer so, als ob das Schlüsselwort COL angegeben würde. Deshalb brauchen Sie diese Schlüsselwörter nicht anzugeben, wenn Sie **IBM Campaign** verwenden.

## Beschreibung

AVG berechnet das arithmetische Mittel oder den Durchschnitt der Zellen im angegebenen Datenbereich. Das arithmetische Mittel wird berechnet, indem der Inhalt aller Zellen addiert und das Ergebnis durch die Anzahl der Zellen geteilt wird. Die Anzahl der von AVG zurückgegebenen Spalten hängt von keyword ab.

- Wenn keyword den Wert ALL hat, gibt AVG eine neue Spalte zurück, die einen Einzelwert (den Durchschnitt aller Zellen in data) enthält.
- Wenn keyword den Wert COL hat, gibt AVG für jede Eingabespalte eine neue Spalte zurück. Jede neue Spalte enthält einen Wert (den Durchschnitt aller Zellen in der entsprechenden Eingabespalte).
- Wenn keyword den Wert ROW hat, gibt AVG eine neue Spalte zurück, die den Durchschnitt für jede Zeile von data enthält.

**Anmerkung:** Leere Zellen werden bei der Berechnung ignoriert.

**Anmerkung:** AVG ist mit der Makrofunktion MEAN identisch.

## Beispiele

TEMP = AVG(V1)
Erstellt eine Spalte namens TEMP, die einen Einzelwert enthält, der das arithmetische Mittel des Inhalts von Spalte V1 darstellt.
TEMP = AVG(V1:V3)
Erstellt eine Spalte namens TEMP, die einen Einzelwert enthält, der das arithmetische Mittel des Inhalts von Spalte V1, V2 und V3 darstellt.
TEMP = AVG(V1[10:20])
Erstellt eine Spalte namens TEMP, die einen Einzelwert enthält, der das arithmetische Mittel der Zellen in den Zeilen 10 bis 20 von Spalte V1 darstellt.
TEMP = AVG(V1[1:5]:V4)
Erstellt eine Spalte namens TEMP, die einen Einzelwert enthält, der das arithmetische Mittel der Zellen in den Zeilen 1 bis 5 der Spalten V1 bis V4 darstellt.

TEMP = AVG(V1:V3, COL)
Erstellt drei neue Spalten namens TEMP, VX und VY. Der Einzelwert in der Spalte TEMP ist das arithmetische Mittel des Inhalts von Spalte V1. Der Einzelwert in der Spalte VX ist das arithmetische Mittel des Inhalts von Spalte V2. Der Einzelwert in der Spalte VY ist das arithmetische Mittel des Inhalts von Spalte V3.
TEMP = AVG(V1[1:5]:V3, COL)
Erstellt drei neue Spalten namens TEMP, VX und VY, die jeweils einen Einzelwert enthalten. Der Wert in der Spalte TEMP ist das arithmetische Mittel der Zellen in den Zeilen 1 bis 5 von Spalte V1. Der Wert in der Spalte VX ist das arithmetische Mittel der Zellen in den Zeilen 1 bis 5 von Spalte V2. Der Wert in der Spalte VY ist das arithmetische Mittel der Zellen in den Zeilen 1 bis 5 von Spalte V3.
TEMP = AVG(V1, ROW)
Erstellt eine Spalte namens TEMP, die dieselben Werte wie Spalte V1 enthält (das arithmetische Mittel einer beliebigen Zahl ist die Zahl selbst).
TEMP = AVG(V1:V3, ROW)
Erstellt eine Spalte namens TEMP, in der jeder Zelleneintrag das arithmetische Mittel der entsprechenden Zeile in den Spalten V1, V2 und V3 darstellt.
TEMP = AVG(V1[1:5]:V3, ROW)
Erstellt eine Spalte namens TEMP, in der die Zellen in den Zeilen 1 bis 5 das arithmetische Mittel der entsprechenden Zeile in den Spalten V1 bis V3 enthalten. Die anderen Zellen in TEMP sind leer.

## Zugehörige Funktionen

Funktion	Beschreibung
SUM oder TOTAL	Berechnet die Summe eines Zellenbereichs.

---

## BETWEEN-Makro

Das BETWEEN-Makro ist nur in IBM Campaign verfügbar.

### Syntax

value1 BETWEEN value2 AND value3

### Parameter

Äquivalent zu value1 >= value2 AND < value3.

### Beschreibung

BETWEEN ist eine spezielle Variante des Vergleichsprädikats. Es kommt auf die Details dieses Prädikats an; die Reihenfolge der Operanden hat einige unerwartete Auswirkungen. Vergleichen Sie dazu die nachfolgenden Beispiele.

**Anmerkung:** FROM und FOR verwenden eine identische Syntax.

## Beispiele

```
10 BETWEEN 5 AND 15 Is true, but: 10 BETWEEN 15 AND 5 Is false:
```

da der entsprechende Ausdruck mit BETWEEN und AND eine bestimmte Reihenfolge aufweist, die keine Rolle spielt, wenn Sie Literale verwenden, aber von großer Bedeutung ist, wenn Sie durch Variablen, Parameter oder sogar Unterabfragen value2 und value3 angeben.

---

## BIT\_AND-Makro

Das BIT\_AND-Makro Makro ist in IBM Campaign und IBM Interact verfügbar.

### Syntax

```
data1 BIT_AND data2 data1 & data2
```

### Parameter

data1

Die nicht negativen Ganzzahlen, die durch bitweises Und mit den Werten in data2 verknüpft werden sollen. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

data2

Die nicht negative(n) Ganzzahl(en), die durch bitweises Und mit den Werten in data1 verknüpft werden soll(en). Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Die Anzahl der Spalten in data2 muss mit der Anzahl der Spalten in data1 übereinstimmen, es sei denn, bei data2 handelt es sich um eine Konstante. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

### Beschreibung

BIT\_AND berechnet das bitweise Und zwischen den beiden angegebenen Datenbereichen. BIT\_AND gibt für jede Eingabespalte eine neue Spalte zurück, wobei jeweils die entsprechende Spalte von data1 durch bitweises Und mit der entsprechenden Spalte von data2 verknüpft wird (d. h., die erste Spalte von data1 wird durch bitweises Und mit der ersten Spalte von data2 verknüpft, die zweite Spalte mit der zweiten Spalte usw.).

Wenn es sich bei data2 um eine Konstante handelt, wird jeder Wert in data1 durch bitweises Und mit dem betreffenden Wert verknüpft. Wenn data2 eine oder mehrere Spalten enthält, werden die Berechnungen zeilenweise zwischen einer Spalte von data1 und einer Spalte von data2 ausgeführt. Die erste Zeile von data1 wird durch bitweises Und mit der ersten Zeile von data2 verknüpft, die zweite Zeile mit der zweiten Zeile usw. Diese zeilenweise ausgeführte Berechnung erzeugt für jede Zeile ein Ergebnis, bis zum letzten Wert der kürzesten Spalte.

**Anmerkung:** Bei dieser Makrofunktion ist die Genauigkeit auf Ganzzahlwerte kleiner als  $2^{24}$  begrenzt. Negative Werte sind nicht zulässig.

**Anmerkung:** Der Operator BIT\_AND kann durch ein Et-Zeichen (&) abgekürzt werden. Mit dem Et-Zeichen können die beiden Argumente getrennt werden (statt z. B. BIT\_AND(V1, 3) anzugeben, können Sie einfach V1&3 eingeben).

## Beispiele

TEMP = 3 BIT_AND 7 oder TEMP = 3 & 7
Erstellt eine neue Spalte TEMP, die den Wert drei enthält (das bitweise Und von 011 und 111 ergibt 011).
TEMP = V1 & 8
Erstellt eine neue Spalte TEMP, in der der Inhalt von Spalte V1 jeweils durch bitweises Und mit dem Binärwert 1000 verknüpft ist.
TEMP = V1 & V1
Erstellt eine neue Spalte TEMP, die denselben Inhalt wie die Spalte V1 enthält (jeder Wert ergibt bei Verknüpfung durch AND mit sich selbst den Wert selbst).
TEMP = V1 & V2
Erstellt eine neue Spalte TEMP, in der jeder Zeilenwert von Spalte V1 durch bitweises Und mit dem entsprechenden Zeilenwert von Spalte V2 verknüpft ist.
TEMP = V1:V3 & V4:V6
Erstellt drei neue Spalten TEMP, VX und VY. In der Spalte TEMP sind die Werte von Spalte V1 durch bitweises Und mit den entsprechenden Zeilenwerten von Spalte V4 verknüpft. In der Spalte VX sind die Werte von Spalte V2 durch bitweises Und mit den Werten von Spalte V5 verknüpft. In der Spalte VY sind die Werte von Spalte V3 durch bitweises Und mit den Werten von Spalte V6 verknüpft.
TEMP = V1[10:20] & V2 oder TEMP = V1[10:20] & V2[1:11]
Erstellt eine neue Spalte TEMP, in deren ersten 11 Zellen die Werte in Zeile 10-20 von Spalte V1 durch bitweises Und mit den Werten in Zeile 1-11 von Spalte V2 verknüpft sind. Die anderen Zellen in TEMP sind leer.

## Zugehörige Funktionen

Funktion	Beschreibung
BIT_NOT	Berechnet das bitweise Nicht des Inhalts des angegebenen Datenbereichs.
BIT_OR	Berechnet das bitweise Oder zwischen den beiden angegebenen Datenbereichen.
BIT_XOR oder XOR	Berechnet das bitweise exklusive Oder zwischen den beiden angegebenen Datenbereichen.

## BIT\_NOT-Makro

Das BIT\_NOT-Makro Makro ist in IBM Campaign und IBM Interact verfügbar.

## Syntax

BIT\_NOT data ~ data

## Parameter

data

Die nicht negativen Ganzzahlen, die durch bitweises Nicht verknüpft werden sollen. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

## Beschreibung

BIT\_NOT gibt das bitweise Nicht der Werte im angegebenen Datenbereich zurück. BIT\_NOT gibt für jede Eingabespalte eine neue Spalte zurück, die das bitweise Nicht der Werte in der entsprechenden Spalte von data enthält.

**Anmerkung:** Bei dieser Makrofunktion ist die Genauigkeit auf Ganzzahlwerte kleiner als  $2^{24}$  begrenzt. Negative Werte sind nicht zulässig.

**Anmerkung:** Wenn eine Spalte in jeder Zeile dieselbe Zahl  $x$  wie data enthält, ist dies dasselbe, als wenn als data die Konstante  $x$  verwendet wird.

**Anmerkung:** Der Operator BIT\_NOT kann durch eine Tilde (~) abgekürzt werden. Stellen Sie die Tilde dem Datenwert voran (statt z. B. BIT\_NOT(V1) anzugeben, können Sie einfach ~V1 eingeben).

## Beispiele

TEMP = BIT_NOT 3 oder TEMP = ~3
Erstellt eine neue Spalte TEMP, die den Wert vier enthält (das bitweise Nicht von 011 ergibt 100).
TEMP = ~V1
Erstellt eine neue Spalte TEMP, in der jeder Wert das bitweise Nicht des Inhalts von Spalte V1 darstellt.
TEMP = ~V1:V3
Erstellt drei neue Spalten TEMP, VX und VY. Die Werte in der Spalte TEMP sind jeweils das bitweise Nicht des Inhalts von Spalte V1, die Werte von Spalte VX sind das bitweise Nicht des Inhalts von Spalte V2 und die Werte von Spalte VY sind das bitweise Nicht des Inhalts von Spalte V3.
TEMP = ~V1[100:200]
Erstellt eine neue Spalte TEMP, in der die ersten 101 Zellen jeweils das bitweise Nicht der Werte in Zeile 100-200 von Spalte V1 darstellen.

## Zugehörige Funktionen

Funktion	Beschreibung
BIT_AND	Berechnet das bitweise Und zwischen den beiden angegebenen Datenbereichen.
BIT_OR	Berechnet das bitweise Oder zwischen den beiden angegebenen Datenbereichen.
BIT_XOR oder XOR	Berechnet das bitweise exklusive Oder zwischen den beiden angegebenen Datenbereichen.

---

## BIT\_OR-Makro

Das BIT\_OR-Makro Makro ist in IBM Campaign und IBM Interact verfügbar.

### Syntax

```
data1 BIT_OR data2 data1 OR data2 data1 | data2
```

### Parameter

data1

Die nicht negativen Ganzzahlen, die durch bitweises Oder mit den Werten in data2 verknüpft werden sollen. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

data2

Die nicht negative(n) Ganzzahl(en), die durch bitweises Oder mit den Werten in data1 verknüpft werden soll(en). Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Die Anzahl der Spalten in data2 muss mit der Anzahl der Spalten in data1 übereinstimmen, es sei denn, bei data2 handelt es sich um eine Konstante. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

### Beschreibung

BIT\_OR berechnet das bitweise Oder zwischen den beiden angegebenen Datenbereichen. BIT\_OR gibt für jede Eingabespalte eine neue Spalte zurück, wobei jeweils die entsprechende Spalte von data1 durch bitweises Oder mit der entsprechenden Spalte von data2 verknüpft wird (d. h., die erste Spalte von data1 wird durch bitweises Oder mit der ersten Spalte von data2 verknüpft, die zweite Spalte mit der zweiten Spalte usw.).

Wenn es sich bei data2 um eine Konstante handelt, wird jeder Wert in data1 durch bitweises Oder mit dem betreffenden Wert verknüpft. Wenn data2 eine oder mehrere Spalten enthält, werden die Berechnungen zeilenweise zwischen einer Spalte von data1 und einer Spalte von data2 ausgeführt. Die erste Zeile von data1 wird durch bitweises Oder mit der ersten Zeile von data2 verknüpft, die zweite Zeile

mit der zweiten Zeile usw. Diese zeilenweise ausgeführte Berechnung erzeugt für jede Zeile ein Ergebnis, bis zum letzten Wert der kürzesten Spalte.

**Anmerkung:** Bei dieser Makrofunktion ist die Genauigkeit auf Ganzzahlwerte kleiner als  $2^{24}$  begrenzt. Negative Werte sind nicht zulässig.

**Anmerkung:** Der Operator BIT\_OR kann durch einen vertikalen Strich (|) abgekürzt werden. Mit dem vertikalen Strich können die beiden Spalten getrennt werden (statt z. B. BIT\_OR(V1, 3)) anzugeben, können Sie einfach V1|3 eingeben. Sie können auch den Ausdruck OR verwenden.

## Beispiele

TEMP = 3 BIT_OR 7 oder TEMP = 3 OR 7 oder TEMP = 3   7
Erstellt eine neue Spalte TEMP, die den Wert sieben enthält (das bitweise Oder von 011 und 111 ergibt 111).
TEMP = V1   8
Erstellt eine neue Spalte TEMP, in der der Inhalt von Spalte V1 jeweils durch bitweises Oder mit dem Binärwert 1000 verknüpft ist.
TEMP = V1   V1
Erstellt eine neue Spalte TEMP, die denselben Inhalt wie die Spalte V1 enthält (jeder Wert ergibt bei Verknüpfung durch OR mit sich selbst den Wert selbst).
TEMP = V1   V2
Erstellt eine neue Spalte TEMP, in der jeder Zeilenwert von Spalte V1 durch bitweises Oder mit dem entsprechenden Zeilenwert von Spalte V2 verknüpft ist.
TEMP = V1:V3   V4:V6
Erstellt drei neue Spalten TEMP, VX und VY. In der Spalte TEMP sind die Werte von Spalte V1 durch logisches Oder mit den entsprechenden Zeilenwerten von Spalte V4 verknüpft. In der Spalte VX sind die Werte von Spalte V2 durch logisches Oder mit den Werten von Spalte V5 verknüpft. In der Spalte VY sind die Werte von Spalte V3 durch logisches Oder mit den Werten von Spalte V6 verknüpft.
TEMP = V1[10:20]   V2 or TEMP = V1[10:20]   V2[1:11]
Erstellt eine neue Spalte TEMP, in deren ersten 11 Zellen die Werte in Zeile 10-20 von Spalte V1 durch bitweises Oder mit den Werten in Zeile 1-11 von Spalte V2 verknüpft sind. Die anderen Zellen in TEMP sind leer.

## Zugehörige Funktionen

Funktion	Beschreibung
BIT_AND	Berechnet das bitweise Und zwischen den beiden angegebenen Datenbereichen.
BIT_NOT	Berechnet das bitweise Nicht des Inhalts des angegebenen Datenbereichs.
BIT_XOR oder XOR	Berechnet das bitweise exklusive Oder zwischen den beiden angegebenen Datenbereichen.

---

## BIT\_XOR-Makro

Das BIT\_XOR-Makro Makro ist in IBM Campaign und IBM Interact verfügbar.

### Syntax

```
data1 BIT_XOR data2
```

### Parameter

data1

Die nicht negativen Ganzzahlen, die durch bitweises exklusives Oder mit den Werten in data2 verknüpft werden sollen. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

data2

Die nicht negative(n) Ganzzahl(en), die durch bitweises exklusives Oder mit den Werten in data1 verknüpft werden soll(en). Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Die Anzahl der Spalten in data2 muss mit der Anzahl der Spalten in data1 übereinstimmen, es sei denn, bei data2 handelt es sich um eine Konstante. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

### Beschreibung

BIT\_XOR berechnet das bitweise exklusive Oder zwischen den beiden angegebenen Datenbereichen. BIT\_XOR gibt für jede Eingabespalte eine neue Spalte zurück, wobei jeweils die entsprechende Spalte von data1 durch bitweises exklusives Oder mit der entsprechenden Spalte von data2 verknüpft wird (d. h., die erste Spalte von data1 wird durch bitweises exklusives Oder mit der ersten Spalte von data2 verknüpft, die zweite Spalte mit der zweiten Spalte usw.).

Wenn es sich bei data2 um eine Konstante handelt, wird jeder Wert in data1 durch bitweises exklusives Oder mit dem betreffenden Wert verknüpft. Wenn data2 eine oder mehrere Spalten enthält, werden die Berechnungen zeilenweise zwischen einer Spalte von data1 und einer Spalte von data2 ausgeführt. Die erste Zeile von data1 wird durch bitweises exklusives Oder mit der ersten Zeile von data2 verknüpft, die zweite Zeile mit der zweiten Zeile usw. Diese zeilenweise ausgeführte Berechnung erzeugt für jede Zeile ein Ergebnis, bis zum letzten Wert der kürzesten Spalte.

**Anmerkung:** Bei dieser Makrofunktion ist die Genauigkeit auf Ganzzahlwerte kleiner als  $2^{24}$  begrenzt. Negative Werte sind nicht zulässig.

## Beispiele

TEMP = 3 BIT_XOR 7
Erstellt eine neue Spalte TEMP, die den Wert vier enthält (das bitweise exklusive Oder von 011 und 111 ergibt 100).
TEMP = V1 BIT_XOR 8
Erstellt eine neue Spalte TEMP, in der der Inhalt von Spalte V1 jeweils durch bitweises exklusives Oder mit dem Binärwert 1000 verknüpft ist.
TEMP = V1 BIT_XOR V1
Erstellt eine neue Spalte TEMP, die nur Nullen enthält (jeder Wert, der durch exklusives Oder mit sich selbst verknüpft wird, ergibt null).
TEMP = V1 BIT_XOR V2
Erstellt eine neue Spalte TEMP, in der jeder Zeilenwert von Spalte V1 durch bitweises exklusives Oder mit dem entsprechenden Zeilenwert von Spalte V2 verknüpft ist.
TEMP = V1:V3 BIT_XOR V4:V6
Erstellt drei neue Spalten TEMP, VX und VY. In der Spalte TEMP sind die Werte von Spalte V1 durch bitweises exklusives Oder mit den entsprechenden Zeilenwerten von Spalte V4 verknüpft. In der Spalte VX sind die Werte von Spalte V2 durch bitweises exklusives Oder mit den Werten von Spalte V5 verknüpft. In der Spalte VY sind die Werte von Spalte V3 durch bitweises exklusives Oder mit den Werten von Spalte V6 verknüpft.
TEMP = V1[10:20] BIT_XOR V2 oder TEMP = V1[10:20] BIT_XOR V2[1:11]
Erstellt eine neue Spalte TEMP, in deren ersten 11 Zellen die Werte in Zeile 10-20 von Spalte V1 durch bitweises exklusives Oder mit den Werten in Zeile 1-11 von Spalte V2 verknüpft sind. Die anderen Zellen in TEMP sind leer.

## Zugehörige Funktionen

Funktion	Beschreibung
BIT_AND	Berechnet das bitweise Und zwischen den beiden angegebenen Datenbereichen.
BIT_NOT	Berechnet das bitweise Nicht des Inhalts des angegebenen Datenbereichs.
BIT_OR	Berechnet das bitweise Oder zwischen den beiden angegebenen Datenbereichen.

---

## CEILING-Makro

Das CEILING-Makro ist nur in IBM Campaign verfügbar.

### Syntax

CEILING(data)

### Parameter

data

Die numerischen Werte, deren Obergrenze berechnet werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Aus-

druck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

## Beschreibung

CEILING berechnet die Obergrenze der Werte im angegebenen Datenbereich. Die Obergrenze einer Zahl ist die kleinste Ganzzahl, die *nicht* kleiner als die Zahl ist. CEILING gibt für jede Eingabespalte eine neue Spalte zurück, die die Obergrenze der Zahlen in der entsprechenden Eingabespalte enthält.

## Beispiele

TEMP = CEILING(4.3)
Erstellt eine neue Spalte TEMP, die den Wert 5 enthält.
TEMP = CEILING(-2.9)
Erstellt eine neue Spalte TEMP, die den Wert -2 enthält.
TEMP = CEILING(V1)
Erstellt eine neue Spalte TEMP, in der jeder Wert die Obergrenze des Inhalts von Spalte V1 darstellt.
TEMP = CEILING(V1:V3)
Erstellt drei neue Spalten TEMP, VX und VY. Die Werte in der Spalte TEMP sind jeweils die Obergrenze des Inhalts von Spalte V1, die Werte von Spalte VX sind die Obergrenze des Inhalts von Spalte V2 und die Werte von Spalte VY sind die Obergrenze des Inhalts von Spalte V3.
TEMP = CEILING(V1[10:20])
Erstellt eine neue Spalte TEMP, in der die ersten 11 Zellen jeweils die Obergrenze der Werte in Zeile 10-20 von Spalte V1 darstellen. Die anderen Zellen in TEMP sind leer.
TEMP = CEILING(V1[50:99]:V2)
Erstellt zwei neue Spalten TEMP und VX, die jeweils Werte in Zeile 1-50 enthalten (die anderen Zellen sind leer). Die Werte in der Spalte TEMP sind jeweils die Obergrenze der Zeilen von Spalte V1, die Werte in der Spalte VX sind die Obergrenze der Zeilen von Spalte V2.

## Zugehörige Funktionen

Funktion	Beschreibung
FLOOR oder INT	Berechnet die Untergrenze jedes Werts im angegebenen Datenbereich.
FRACTION	Berechnet die Nachkommastellen jedes Werts im angegebenen Datenbereich.
TRUNCATE	Berechnet den Ganzzahlanteil jedes Werts im angegebenen Datenbereich.

## COLUMN-Makro

Das COLUMN-Makro Makro ist nur in IBM Campaign verfügbar.

## Syntax

COLUMN(data [, data]...) or (data [, data]...)

## Parameter

data

Ein Wert, der beim Erstellen einer Spalte verwendet werden soll. Dabei kann es sich um einen konstanten Wert (numerisch oder ASCII-Text in Anführungszeichen), eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Dieser Parameter kann mehrmals wiederholt werden, doch nachfolgende Parameter müssen dieselbe Dimensionalität (d. h. Spaltenbreite) wie der erste Parameter aufweisen. Alle Werte in allen data-Parametern müssen entweder numerisch oder ASCII-Text sein (d. h., Zahlen- und Textwerte dürfen nicht gemischt werden). Wenn mehrere data-Parameter angegeben werden, müssen alle dieselbe Anzahl von Spalten aufweisen. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

## Beschreibung

COLUMN verkettet die Eingaben vertikal zu Spalten einer Funktionsgruppe. Dabei wird dieselbe Anzahl neuer Spalten zurückgegeben, wie im jeweiligen Eingabeparameter angegeben. Es kann eine unbegrenzte Zahl von Argumenten angegeben werden. Alle Argumente müssen entweder numerisch oder ASCII-Zeichenfolgen sein (d. h., Zahlen- und Textwerte dürfen nicht gemischt werden).

**Anmerkung:** Die Makrofunktion COLUMN kann abgekürzt werden, indem die data-Argumente innerhalb runder Klammern durch Kommas getrennt aufgelistet werden (z. B. TEMP = MEAN((1,2,3,4), ALL)). Wenn es nicht um den Einsatz innerhalb einer anderen Makrofunktion geht, sind die Klammern nicht erforderlich; beispielsweise ist V1=1,2,3 äquivalent zu V1=COLUMN(1,2,3).

## Beispiele

TEMP = COLUMN(3, 4, 5) or TEMP = (3,4,5) or TEMP = 3,4,5
Erstellt eine neue Spalte TEMP, deren erste drei Zellen die Werte 3, 4 und 5 enthalten.
TEMP = COLUMN("one", "two", "three")
Erstellt eine neue Spalte TEMP, deren erste drei Zellen die Werte eins, zwei und drei enthalten.
TEMP = AVG(V1), STDV(V1)
Erstellt eine neue Spalte TEMP, die in der ersten Zelle den Durchschnitt von Spalte V1 und in der zweiten Zelle die Standardabweichung von Spalte V1 enthält.
TEMP = V1:V2, V3:V4
Erstellt zwei neue Spalten TEMP und VX. Die Spalte TEMP enthält die Werte aus Spalte V1, gefolgt von den Werten aus Spalte V3. Die Spalte VX enthält die Werte aus Spalte V2, gefolgt von den Werten aus Spalte V4.
TEMP = V1:V2, V3:V4
Erstellt zwei neue Spalten TEMP und VX. Die Spalte TEMP enthält die Werte aus Zelle 1-10 von Spalte V1, gefolgt von allen Werten aus Spalte V3. Die Spalte VX enthält die Werte aus Zelle 1-10 von Spalte V2, gefolgt von allen Werten aus Spalte V4.

TEMP = V1:V2, V3:V4

Erstellt zwei neue Spalten TEMP und VX, die jeweils einen Einzelwert enthalten. Die Spalte TEMP enthält den Durchschnitt der Spalten V1 und V2. Die Spalte VX enthält den Durchschnitt der Spalten V3 und V4.

---

## COS-Makro

Das COS-Makro Makro ist nur in IBM Campaign verfügbar.

### Syntax

`COS(data [, units_keyword])`

### Parameter

`data`

Die numerischen Werte, deren Kosinus berechnet werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

`units_keyword`

Dieses optionale Schlüsselwort legt fest, ob die Eingabewerte und Ergebnisse als Grad oder als Radiant interpretiert werden. Wählen Sie eine der folgenden Angaben aus:

RADIAN - Führt die Berechnungen in Radiant aus (Standard)

DEGREE - Führt die Berechnungen in Grad aus

Wird dieser Parameter nicht angegeben, ist der Standard Radiant. (Die Umrechnung von Radiant in Grad erfolgt durch Division durch  $\pi$  und Multiplikation mit 180.)

Weitere Informationen zur Verwendung von Schlüsselwörtern in IBM Campaign finden Sie in „Formatspezifikationen“ auf Seite 9.

### Beschreibung

COS berechnet den Kosinus der Werte im angegebenen Datenbereich. COS gibt für jede Eingabespalte eine neue Spalte zurück, die den Kosinus der Zahlen in der entsprechenden Eingabespalte enthält.

### Beispiele

TEMP = COS(PI) oder TEMP = COS(PI, 0) oder TEMP = COS(PI, RADIAN)

Gibt eine neue Spalte TEMP zurück, die den Einzelwert -1 enthält.

TEMP = COS(90, 1) oder TEMP = COS(90, DEGREE)

Gibt eine neue Spalte TEMP zurück, die den Einzelwert null enthält.

TEMP = COS(V1) oder TEMP = COS(V1, 0) oder TEMP = COS(V1, RADIAN)
Erstellt eine neue Spalte TEMP, in der jeder Wert den Kosinus (in Radiant) des Inhalts von Spalte V1 darstellt.
TEMP = COS(V1:V3, 1)
Erstellt drei neue Spalten TEMP, VX und VY. Die Werte in der Spalte TEMP sind jeweils der Kosinus des Inhalts von Spalte V1, die Werte von Spalte VX sind der Kosinus des Inhalts von Spalte V2 und die Werte von Spalte VY sind der Kosinus des Inhalts von Spalte V3. Alle Werte sind in Grad angegeben.
TEMP = COS(V1[10:20])
Erstellt eine neue Spalte TEMP, in der die ersten 11 Zellen jeweils den Kosinus der Werte in Zeile 10-20 von Spalte V1 darstellen. Die anderen Zellen in TEMP sind leer.
TEMP = COS(V1[1:5]:V2)
Erstellt zwei neue Spalten TEMP und VX, die jeweils die Werte in Zeile 1-5 enthalten (die anderen Zellen sind leer). Die Werte in der Spalte TEMP sind der Kosinus der entsprechenden Zeilen von Spalte V1, die Werte in der Spalte VX sind der Kosinus der entsprechenden Zeilen von Spalte V2. Alle Werte sind in Radiant angegeben.

## Zugehörige Funktionen

Funktion	Beschreibung
ACOS	Berechnet den Arkuskosinus des Inhalts des angegebenen Datenbereichs.
COSH	Berechnet den Hyperbelkosinus des Inhalts des angegebenen Datenbereichs.
SIN	Berechnet den Sinus des Inhalts des angegebenen Datenbereichs.
TAN	Berechnet den Tangens des Inhalts des angegebenen Datenbereichs.

---

## COSH-Makro

Das COSH-Makro Makro ist nur in IBM Campaign verfügbar.

### Syntax

COSH(data [, units\_keyword])

### Parameter

data

Die numerischen Werte, deren Hyperbelkosinus berechnet werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

units\_keyword

Dieses optionale Schlüsselwort legt fest, ob die Eingabewerte und Ergebnisse als Grad oder als Radiant interpretiert werden. Wählen Sie eine der folgenden Angaben aus:

RADIAN - Führt die Berechnungen in Radiant aus (Standard)

DEGREE - Führt die Berechnungen in Grad aus

Wird dieser Parameter nicht angegeben, ist der Standard Radiant. (Die Umrechnung von Radiant in Grad erfolgt durch Division durch PI und Multiplikation mit 180.)

Weitere Informationen zur Verwendung von Schlüsselwörtern in IBM Campaign finden Sie in „Formatspezifikationen“ auf Seite 9.

## Beschreibung

COSH berechnet den Hyperbelkosinus der Werte im angegebenen Datenbereich. Bei  $x$  in Radiant ist der Hyperbelkosinus einer Zahl

$$\cosh(x) = \frac{e^x + e^{-x}}{2}$$

Dabei ist  $e$  die natürliche Zahl, 2,7182818. COSH gibt für jede Eingabespalte eine neue Spalte zurück, die den Hyperbelkosinus der Zahlen in der entsprechenden Eingabespalte enthält.

**Anmerkung:** Wenn der Wert  $x$  zu groß ist, wird ein Überlauferfehler zurückgegeben. Dies geschieht, wenn  $\cosh(x)$  den maximalen 32-Bit-Gleitkommawert überschreitet.

## Beispiele

TEMP = COSH(0) oder TEMP = COSH(0, 0) oder TEMP = COSH(0, RADIAN)
Gibt eine neue Spalte TEMP zurück, die den Wert eins enthält.
TEMP = COSH(V1)
Erstellt eine neue Spalte TEMP, in der jeder Wert den Hyperbelkosinus (in Radiant) des Inhalts von Spalte V1 darstellt.
TEMP = COSH(V1:V3, 1) oder TEMP = COSH(V1:V3, DEGREE)
Erstellt drei neue Spalten TEMP, VX und VY. Die Werte in der Spalte TEMP sind jeweils der Hyperbelkosinus des Inhalts von Spalte V1, die Werte von Spalte VX sind der Hyperbelkosinus des Inhalts von Spalte V2 und die Werte von Spalte VY sind der Hyperbelkosinus des Inhalts von Spalte V3. Alle Werte sind in Grad angegeben.
TEMP = COSH(V1[10:20])
Erstellt eine neue Spalte TEMP, in der die ersten 11 Zellen jeweils den Hyperbelkosinus der Werte in Zeile 10-20 von Spalte V1 darstellen. Die anderen Zellen in TEMP sind leer.

TEMP = COSH(V1[1:5]:V2)

Erstellt zwei neue Spalten TEMP und VX, die jeweils die Werte in Zeile 1-5 enthalten (die anderen Zellen sind leer). Die Werte in der Spalte TEMP sind der Hyperbelkosinus der entsprechenden Zeilen von Spalte V1, die Werte in der Spalte VX sind der Hyperbelkosinus der entsprechenden Zeilen von Spalte V2. Alle Werte sind in Radiant angegeben.

## Zugehörige Funktionen

Funktion	Beschreibung
ACOS	Berechnet den Arkuskosinus des Inhalts des angegebenen Datenbereichs.
COS	Berechnet den Kosinus des Inhalts des angegebenen Datenbereichs.
SINH	Berechnet den Hyperbelsinus des Inhalts des angegebenen Datenbereichs.
TANH	Berechnet den Hyperbeltangens des Inhalts des angegebenen Datenbereichs.

---

## COT-Makro

Das COT-Makro Makro ist nur in IBM Campaign verfügbar.

### Syntax

COT(data [, units\_keyword])

### Parameter

data

Die numerischen Werte, deren Kotangens berechnet werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

units\_keyword

Dieses optionale Schlüsselwort legt fest, ob die Eingabewerte und Ergebnisse als Grad oder als Radiant interpretiert werden. Wählen Sie eine der folgenden Angaben aus:

RADIAN - Führt die Berechnungen in Radiant aus (Standard)

DEGREE - Führt die Berechnungen in Grad aus

Wird dieser Parameter nicht angegeben, ist der Standard Radiant. (Die Umrechnung von Radiant in Grad erfolgt durch Division durch  $\pi$  und Multiplikation mit 180.)

Weitere Informationen zur Verwendung von Schlüsselwörtern in IBM Campaign finden Sie in „Formatspezifikationen“ auf Seite 9.

## Beschreibung

COT berechnet den Kotangens der Werte im angegebenen Datenbereich. Der Kotangens ist der Kehrwert des Tangens. COT gibt für jede Eingabespalte eine neue Spalte zurück, die den Kotangens der Zahlen in der entsprechenden Eingabespalte enthält.

**Anmerkung:** Wenn eine Zelle einen Wert enthält, dessen Tangens null ist, ist der Kotangens unendlich. In diesem Fall gibt COT die größte 32-Bit-Gleitkommazahl zurück.

## Beispiele

TEMP = COT(90) oder TEMP = COT(90, 0) oder TEMP = COT(90, RADIAN) Gibt eine neue Spalte TEMP zurück, die den Wert -0.5 enthält.
TEMP = COT(0) Gibt eine neue Spalte TEMP zurück, die den Wert MAX_FLOAT_32 enthält.
TEMP = COT(V1, 1) oder TEMP = COT(V1, DEGREE) Erstellt eine neue Spalte TEMP, in der jeder Wert den Kotangens des Inhalts (in Grad) von Spalte V1 darstellt.
TEMP = COT(V1:V3, 1) Erstellt drei neue Spalten TEMP, VX und VY. Die Werte in der Spalte TEMP sind jeweils der Kotangens des Inhalts von Spalte V1, die Werte von Spalte VX sind der Kotangens des Inhalts von Spalte V2 und die Werte von Spalte VY sind der Kotangens des Inhalts von Spalte V3. Alle Werte sind in Grad angegeben.
TEMP = COT(V1[10:20]) Erstellt eine neue Spalte TEMP, in der die ersten 11 Zellen jeweils den Kotangens der Werte in Zeile 10-20 von Spalte V1 darstellen. Die anderen Zellen in TEMP sind leer.
TEMP = COT(V1[1:5]:V2) Erstellt zwei neue Spalten TEMP und VX, die jeweils die Werte in Zeile 1-5 enthalten (die anderen Zellen sind leer). Die Werte in der Spalte TEMP sind der Kotangens der entsprechenden Zeilen von Spalte V1, die Werte in der Spalte VX sind der Kotangens der entsprechenden Zeilen von Spalte V2. Alle Werte sind in Radiant angegeben.

## Zugehörige Funktionen

Funktion	Beschreibung
ACOT	Berechnet den Arkuskotangens des Inhalts des angegebenen Datenbereichs.
COS	Berechnet den Kosinus des Inhalts des angegebenen Datenbereichs.
SIN	Berechnet den Sinus des Inhalts des angegebenen Datenbereichs.
TAN	Berechnet den Tangens des Inhalts des angegebenen Datenbereichs.

---

## COUNT-Makro

Das COUNT-Makro Makro ist nur in IBM Campaign verfügbar.

## Syntax

COUNT(data)

## Parameter

data

Der Zellenbereich, in dem die Anzahl der Zellen gezählt werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

## Beschreibung

COUNT berechnet die Anzahl der Werte im angegebenen Datenbereich. COUNT gibt eine neue Spalte mit einem Einzelwert zurück, der die Anzahl der Zellen, die Werte enthalten, innerhalb des angegebenen Datenbereichs darstellt.

**Anmerkung:** Bei einer leeren Spalte wird null zurückgegeben.

## Beispiele

TEMP = COUNT(AVG(V1:V5))
Erstellt eine neue Spalte TEMP, die den Einzelwert eins enthält (die Funktion AVG gibt im Standardmodus eine einzelne Zelle zurück).
TEMP = COUNT(V1)
Erstellt eine neue Spalte TEMP, die einen Einzelwert enthält, der die Anzahl der Zellen mit Werten in Spalte V1 angibt.
TEMP = COUNT(V1:V3)
Erstellt eine neue Spalte TEMP, die einen Einzelwert enthält, der die Anzahl der Zellen mit Werten in den Spalten V1, V2 und V3 angibt.
TEMP = COUNT(V1[10:20])
Erstellt eine neue Spalte TEMP, die den Wert 11 enthält (Bereiche schließen die Grenzen ein), sofern sämtliche Zellen Werte enthalten.
TEMP = COUNT(V1[1:5]:V4)
Erstellt eine neue Spalte TEMP, die den Wert 20 enthält (5 Zellen in jeder Spalte bei 4 Spalten ergeben 20 Zellen), sofern sämtliche Zellen Werte enthalten.
TEMP = COUNT(V1[1:10])
Erstellt eine neue Spalte TEMP, die den Wert 3 enthält, sofern die Zeilen 1-3 von Spalte V1 Werte enthalten und die Zeilen 4-10 leer sind.

## Zugehörige Funktionen

Funktion	Beschreibung
SUM oder TOTAL	Berechnet die Summe eines Zellenbereichs.

---

## CURRENT\_DATE-Makro

Das CURRENT\_DATE-Makro Makro ist in IBM Campaign und IBM Interact verfügbar.

### Syntax

```
CURRENT_DATE([format])
```

### Parameter

format

Eines der Schlüsselwörter in der folgenden Tabelle, das das Datumsformat von date\_string angibt.

**Anmerkung:** Weitere Informationen zu den verfügbaren Datumsformaten finden Sie unter "Gültige Formatschlüsselwörter".

### Beschreibung

CURRENT\_DATE gibt das aktuelle Datum in format zurück. Das Datum wird durch die Systemzeit auf dem IBM Server festgelegt. Wenn kein Schlüsselwort format angegeben wird, wird standardmäßig das Format DELIM\_M\_D\_Y verwendet.

Bei allen empfohlenen Datenbanken versucht IBM Campaign, das Makro CURRENT\_DATE mithilfe eines von der Datenbank unterstützten SQL-Aufrufs für die aktuelle Uhrzeit (z. B. SYSDATE, GETDATE, DATE oder TODAY) in der Datenbank auszuführen. In diesen Fällen werden von dieser Makrofunktion alle Parameter (einschließlich des Datumsformats) ignoriert und die Ausgabe schließt alle Angaben ein, die von der Datenbank zurückgegeben werden (unter Umständen kann die Ausgabe also eine Zeitkomponente enthalten). Falls dies der Fall ist und nur das Datum oder das Datum in einem anderen Format zurückgegeben werden soll, können Sie mit direktem SQL ein benutzerdefiniertes Makro schreiben oder IBM Makros verwenden. Beispiel:

```
DATE_STRING(CURRENT_JULIAN( ), ...)
```

In einigen Fällen wird auf dem IBM Campaign-Server das Makro CURRENT\_DATE() ausgeführt (nämlich wenn eine unstrukturierte Datei oder eine nicht empfohlene Datenbank ohne entsprechende SQL-Unterstützung vorliegt oder wenn der Campaign-Makroausdruck in der Datenbank nicht aufgelöst werden kann). In diesen Fällen werden alle Parameter erkannt und die Ausgabe wird im ausgewählten Format zurückgegeben.

**Anmerkung:** Nicht alle Formate, die in IBM Campaign verfügbar sind, werden von IBM Interact unterstützt.

Beachten Sie, dass Sie möglicherweise das Makro DATE\_FORMAT verwenden müssen, um das CURRENT\_DATE für Ihren Datenbanktyp anzupassen. Zum Beispiel kann bei DB2 das folgende Makro verwendet werden:

```
table_name = CURRENT_DATE()-1
```

Bei Oracle müssen Sie das Makro DATE\_FORMAT jedoch wie folgt verwenden:

```
table_name = DATE_FORMAT(CURRENT_DATE()-1, DELIM_M_D_YYYY, '%Y-%m-%d')
```

## Beispiele

Wenn das aktuelle Datum der 13. September 2015 ist, gibt `CURRENT_DATE()` "09/13/15" zurück.

## Zugehörige Funktionen

Funktion	Beschreibung
<code>DATE_FORMAT</code>	Konvertiert Datumsangaben von einem Format in ein anderes.
<code>DATE_JULIAN</code>	Gibt das julianische Datum der Eingabe zurück.
<code>DATE_STRING</code>	Gibt die Datumszeichenfolge des julianischen Datums zurück.
<code>DATE</code>	Konvertiert eine Datumszeichenfolge in ein julianisches Datum.

---

## CURRENT\_DAY-Makro

Das `CURRENT_DAY`-Makro ist in IBM Campaign und IBM Interact verfügbar.

### Syntax

`CURRENT_DAY()`

### Beschreibung

`CURRENT_DAY` gibt den aktuellen Tag des Monats als Zahl zwischen 1 und 31 zurück. Das Datum wird durch die Systemzeit auf dem IBM Server festgelegt.

### Beispiele

Wenn das aktuelle Datum der 19. Juni ist, gibt `CURRENT_DAY()` die Zahl 19 zurück.

## Zugehörige Funktionen

Funktion	Beschreibung
<code>CURRENT_JULIAN</code>	Gibt die julianische Zahl für das aktuelle Datum zurück.
<code>CURRENT_MONTH</code>	Gibt den aktuellen Monat als Zahl zurück.
<code>CURRENT_TIME</code>	Gibt die aktuelle Uhrzeit als Zeichenfolge zurück.
<code>CURRENT_WEEKDAY</code>	Gibt den aktuellen Wochentag als Zahl zurück.
<code>CURRENT_YEAR</code>	Gibt das aktuelle Jahr als Zahl zurück.

---

## CURRENT\_JULIAN-Makro

Das `CURRENT_JULIAN`-Makro ist nur in IBM Campaign verfügbar.

## Syntax

CURRENT\_JULIAN()

## Beschreibung

CURRENT\_JULIAN() gibt die julianische Zahl für das aktuelle Datum (die Anzahl der seit dem 1. Januar 0000 vergangenen Tage) zurück. Dies ist äquivalent zu dem Makro DATE(CURRENT\_DATE()).

## Beispiele

Wenn das aktuelle Datum der 31. August 2000 ist, gibt CURRENT\_JULIAN() die Zahl 730729 zurück.

## Zugehörige Funktionen

Funktion	Beschreibung
CURRENT_DAY	Gibt den aktuellen Tag als Zahl zurück.
CURRENT_MONTH	Gibt den aktuellen Monat als Zahl zurück.
CURRENT_TIME	Gibt die aktuelle Uhrzeit als Zeichenfolge zurück.
CURRENT_WEEKDAY	Gibt den aktuellen Wochentag als Zahl zurück.
CURRENT_YEAR	Gibt das aktuelle Jahr als Zahl zurück.

---

## CURRENT\_MONTH-Makro

Das CURRENT\_MONTH-Makro Makro ist in IBM Campaign und IBM Interact verfügbar.

## Syntax

CURRENT\_MONTH()

## Beschreibung

CURRENT\_MONTH gibt den aktuellen Monat des Jahres als Zahl zwischen 1 und 12 zurück.

## Beispiele

Wenn das aktuelle Datum der 19. Juni ist, gibt CURRENT\_MONTH() die Zahl 6 zurück.

## Zugehörige Funktionen

Funktion	Beschreibung
CURRENT_DAY	Gibt den aktuellen Tag als Zahl zurück.
CURRENT_JULIAN	Gibt die aktuelle julianische Zahl zurück.
CURRENT_TIME	Gibt die aktuelle Uhrzeit als Zeichenfolge zurück.
CURRENT_WEEKDAY	Gibt den aktuellen Wochentag als Zahl zurück.

Funktion	Beschreibung
CURRENT_YEAR	Gibt das aktuelle Jahr als Zahl zurück.

---

## CURRENT\_TIME-Makro

Das CURRENT\_TIME-Makro Makro ist nur in IBM Campaign verfügbar.

### Syntax

CURRENT\_TIME()

### Beschreibung

CURRENT\_TIME gibt die aktuelle Uhrzeit als Zeichenfolge zurück. Die Uhrzeit wird durch die Systemzeit auf dem IBM Server festgelegt.

## Datumseinstellung in Ihrer Webanwendung

Damit Datumsangaben in Ihrer Webanwendung bei aktuellen Versionen von IBM Campaign korrekt angezeigt werden, muss die Konfigurationsdatei Ihres Back-End-Servers ordnungsgemäß konfiguriert werden. Dies ist besonders wichtig bei den Parametern `dDateFormat` und `DateOutputFormatString` für die Datenbank, die die Systemtabellen enthält. Wenn diese nicht ordnungsgemäß konfiguriert sind, werden Datumsangaben auch in Campaign falsch angezeigt. Diese Eigenschaften können mithilfe von IBM Marketing Platform konfiguriert werden.

### Sprachspezifische Datumsangaben in Ihrer Webanwendung

**Anmerkung:** Alle referenzierten Dateien werden vom Installationsprogramm der Webanwendung installiert, sofern nicht anders angegeben.

**Wichtig:** `webapphome` steht für das Verzeichnis, in dem die Campaign-Webanwendung installiert wurde. `language_code` steht für die Spracheinstellung(en), die Sie für Ihr System festlegen.

1. Bearbeiten Sie die Datei `webapphome/conf/campaign_config.xml`, um sicherzustellen, dass `language_code` in der durch Kommas getrennten Liste im Tag `<supportedLocales>` enthalten ist, wie im Folgenden gezeigt:  
`<supportedLocales>en_US, language_code</supportedLocales>`
2. Kopieren Sie im Verzeichnis `webapphome/webapp` die gesamte Verzeichnisstruktur von `en_US` nach `language_code` (Groß-/Kleinschreibung muss beachtet werden).
3. Kopieren Sie in `webapphome/webapp/WEB-INF/classes/resources` `StaticMessages_en_US.properties` nach `StaticMessages_language_code.properties`. Kopieren Sie außerdem `ErrorMessages_en_US.properties` nach `ErrorMessages_language_code.properties`.
4. Bearbeiten Sie `StaticMessages_language_code.properties`: Suchen Sie nach `DatePattern` und ändern Sie den Eintrag in `DatePattern=dd/MM/yyyy` (Groß-/Kleinschreibung muss beachtet werden).

**Anmerkung:** Dieses Format wird von Java<sup>™</sup> definiert. Vollständige Informationen zu dem Format finden Sie in der Java-Dokumentation für `java.text.SimpleDateFormat` unter <http://java.sun.com>. Die Datei `StaticMessages.properties` braucht nicht geändert zu werden.

5. Bei WebSphere: Erstellen Sie eine neue JAR-Datei der Webanwendung.

6. Bei WebLogic: Entfernen Sie das aktuelle Modul der Webanwendung.
  - a. Fügen Sie das neue Modul hinzu.
  - b. Führen Sie die Bereitstellung der Webanwendung erneut aus.
  - c. Ein Neustart des Campaign-Listeners ist nicht erforderlich.
7. Stellen Sie sicher, dass `language_code` in der Spracheinstellung des Web-Browsers die höchste Priorität hat. Weitere Informationen finden Sie in den folgenden Abschnitten, die die korrekte Spracheinstellung für Ihren Web-Browser und die Anzeige einer bestimmten Sprache auf Ihrem Computer behandeln.

**Anmerkung:** Achten Sie darauf, in `language_code` einen Bindestrich und keinen Unterstrich zu verwenden. Die Konfiguration der Webanwendung ist der einzige Ort, an dem anstelle eines Unterstrichs ein Bindestrich verwendet wird.

8. Melden Sie sich bei Campaign an. Datumsangaben sollten in Campaign in dem Format angezeigt werden, das Sie in `StaticMessages_language_code.properties` angegeben haben.

Informationen zur Konfiguration der Uhrzeit für IBM Campaign finden Sie in der Dokumentation zu *IBM Campaign*.

### Beispiele

Wenn die Uhrzeit 10:54 ist, gibt `CURRENT_TIME()` die Zeichenfolge "10:54:00" zurück.

### Zugehörige Funktionen

Funktion	Beschreibung
<code>CURRENT_DAY</code>	Gibt den aktuellen Tag als Zahl zurück.
<code>CURRENT_JULIAN</code>	Gibt die aktuelle julianische Zahl zurück.
<code>CURRENT_WEEKDAY</code>	Gibt den aktuellen Wochentag als Zahl zurück.
<code>CURRENT_YEAR</code>	Gibt das aktuelle Jahr als Zahl zurück.

---

## CURRENT\_WEEKDAY-Makro

Das `CURRENT_WEEKDAY`-Makro ist in IBM Campaign und IBM Interact verfügbar.

### Syntax

`CURRENT_WEEKDAY()`

### Beschreibung

`CURRENT_WEEKDAY` gibt den aktuellen Wochentag als Zahl zwischen 0 und 6 zurück. Der Sonntag wird als 0 dargestellt, der Montag als 1 usw.

### Beispiele

Wenn der aktuelle Wochentag ein Freitag ist, gibt `CURRENT_WEEKDAY()` die Zahl 5 zurück.

## Zugehörige Funktionen

Funktion	Beschreibung
CURRENT_DAY	Gibt den aktuellen Tag als Zahl zurück.
CURRENT_JULIAN	Gibt die aktuelle julianische Zahl zurück.
CURRENT_MONTH	Gibt den aktuellen Monat als Zahl zurück.
CURRENT_TIME	Gibt die aktuelle Uhrzeit als Zeichenfolge zurück.
CURRENT_YEAR	Gibt das aktuelle Jahr als Zahl zurück.

---

## CURRENT\_YEAR-Makro

Das CURRENT\_YEAR-Makro Makro ist in IBM Campaign und IBM Interact verfügbar.

### Syntax

CURRENT\_YEAR()

### Beschreibung

CURRENT\_YEAR gibt das aktuelle Jahr als Zahl zurück.

### Beispiele

Wenn das aktuelle Jahr 2000 ist, gibt CURRENT\_YEAR() die Zahl 2000 zurück.

## Zugehörige Funktionen

Funktion	Beschreibung
CURRENT_DAY	Gibt den aktuellen Tag als Zahl zurück.
CURRENT_JULIAN	Gibt die aktuelle julianische Zahl zurück.
CURRENT_MONTH	Gibt den aktuellen Monat als Zahl zurück.
CURRENT_TIME	Gibt die aktuelle Uhrzeit als Zeichenfolge zurück.
CURRENT_WEEKDAY	Gibt den aktuellen Wochentag als Zahl zurück.
MONTHOF	Gibt den Monat des Jahres als Zahl zurück.
WEEKDAYOF	Gibt den Wochentag der Woche als Zahl zurück.
YEAROF	Gibt das Jahr als Zahl zurück.

---

## DATE

### Syntax

DATE(input\_date, [input\_date format])

## Parameter

date\_string

Eine Textzeichenfolge, die ein gültiges Datum darstellt.

format

Eines der Schlüsselwörter in der Tabelle unter "Gültige Datumsformatschlüsselwörter", das das Datumsformat von date\_string angibt (optional).

## Beschreibung

Das DATE-Makro von Interact konvertiert ein Eingabedatum in einen formatneutralen Ganzzahlwert.

Das DATE-Makro wird wie folgt berechnet:  $DATE(X) = 365 +$  die Anzahl ganzer verstrichener Tage seit Mittag am 1. Januar n. Chr. Für DATE (X) kann ein optionales Schlüsselwort für das Datumseingabeformat bereitgestellt werden, um anzugeben, wie die DATE-Eingabe geparkt werden soll. Wenn kein Schlüsselwort format angegeben wird, wird standardmäßig das Format DELIM\_M\_D\_Y verwendet. Weitere Informationen zu gültigen Datumsformaten finden Sie unter Gültige Datumsformatschlüsselwörter.

Datumsformate weisen entweder eine feste Breite (dann wird der 28. Februar 1970 im Format MMDDYYYY als 02281970 dargestellt) oder einen Begrenzer auf (dann kann der 28. Februar 1970 im Format DELIM\_M\_D\_YY etwa als 2-28-1970 oder 02/28/1970 dargestellt werden).

In begrenzten Formaten kann der Begrenzer ein Schrägstrich (/), ein Strich (-), ein Leerzeichen ( ), ein Komma (,) oder ein Doppelpunkt (:) sein; Jahre können 2- oder 4-stellig dargestellt werden; Monate können ausgeschrieben (z. B. Februar), abgekürzt (z. B. Feb) oder numerisch dargestellt werden (z. B. 2 oder 02).

Für alle zweistellig angegebenen Jahre gilt Folgendes:

- Interact setzt standardmäßig voraus, dass zweistellige Daten mit Begrenzern zwischen den Jahren 1920-2020 liegen.
- Zweistellige Jahresangaben, die kleiner als der Grenzwert für das Millennium sind (standardmäßig 20, kann jedoch vom JVM-Parameter definiert werden), werden dem 21. Jahrhundert zugerechnet.
- Zweistellige Jahresangaben, die größer-gleich dem Grenzwert sind, werden dem 20. Jahrhundert zugerechnet.

### Anmerkung:

- Nicht alle DATE-Formate, die in Campaign verfügbar sind, werden von Interact unterstützt.
- Weitere Informationen zu zweistelligen Jahren finden Sie in Gültige Datumsformatschlüsselwörter.
- Weitere Informationen zum Konfigurieren des Grenzwerts für das Millennium bei zweistelligen Jahren finden Sie im Abschnitt "JVM-Argumente" im Interact Optimierungshandbuch.

Dieses Makro ist in IBM Interact verfügbar.

Viele geschäftskritische Systeme verwenden julianische Datumsoffsets. Das Ergebnis des DATE()-Makros von Interact bezieht sich wie folgt auf das julianische Datum:

Julianisches Datum = DATE(...) + 1,721,059 + Teil des Tages, der seit dem vorherigen Mittag verstrichen ist.

Nützliche DATE()-Werte für den Zeitraum n. Chr. :

- January 1, 2050 A.D gibt 748,749 zurück.
- January 1, 2000 A.D gibt 730,486 zurück.
- January 1, 1990 A.D gibt 726,834 zurück.
- January 1, 1900 A.D gibt 693,962 zurück.
- January 1, 0001 A.D gibt 365 zurück.

**Anmerkung:** In Übereinstimmung mit der ISO-Norm 8601 und der XML-Schemadefinition für Date- und DateTime-Objekte wird der proleptische gregorianische Kalender zum Berechnen der Anzahl verstrichener Tage verwendet. In diesem Kalendersystem entspricht das hypothetische Jahr 0000 n. Chr. dem Jahr 0001 v. Chr.

## Beispiele

DATE("8/31/2000") gibt die Zahl 730,729 zurück.

DATE("8/31/2000",DELIM\_MM\_DD\_YYYY) gibt die Zahl 730,729 zurück.

DATE("2015-01-01",DELIM\_Y\_M\_D) gibt die Zahl 735,965 zurück.

DATE("01",DD), DATE("0101",MDDD) und DATE("1970-01-01",DELIM\_Y\_M\_D) gibt die Zahl 719,529 zurück.

## Zugehörige Funktionen

Funktion	Beschreibung
DATE_FORMAT	Konvertiert Datumsangaben von einem Format in ein anderes.
DATE_JULIAN	Gibt das julianische Datum der Eingabe zurück.
DATE_STRING	Gibt die Datumszeichenfolge des julianischen Datums zurück.
CURRENT_DATE	Gibt das aktuelle Datum in einem bestimmten Format zurück.

---

## DATE\_FORMAT-Makro

Das DATE\_FORMAT-Makro Makro ist in IBM Campaign und IBM Interact verfügbar.

### Syntax

DATE\_FORMAT(date\_string, input\_format, output\_format)

## Parameter

date\_string

Ein Text, der ein gültiges Datum darstellt.

input\_format

Eines der Schlüsselwörter in der folgenden Tabelle, das das Datumsformat von date\_string angibt.

output\_format

Eines der Schlüsselwörter in der folgenden Tabelle, das das vorgesehene Ausgabeformatsformat angibt.

## Beschreibung

DATE\_FORMAT() wandelt ein Datum von input\_format in ein anderes output\_format um.

Wenn das Datum eine feste Breite aufweist, muss dem Format einer der folgenden Werte zugewiesen werden:

- DDMMYY[YY]
- DDMMMYY[YY]
- MMDDYY[YY]
- MMMDDYY[YY]
- YY[YY]MMDD
- YY[YY]MMMDD

MM ist eine aus 2 Ziffern, MMM eine aus 3 Buchstaben bestehende Abkürzung des Monats.

Wenn das Datum Begrenzer aufweist (als Begrenzer können Leerzeichen, Gedankenstriche und Schrägstriche verwendet werden), muss dem Format einer der folgenden Werte zugewiesen werden:

- DELIM\_D\_M\_Y
- DELIM\_M\_D\_Y
- DELIM\_Y\_M\_D

**Anmerkung:** Nicht alle Formate, die in IBM Campaign verfügbar sind, werden von IBM Interact unterstützt.

## Beispiele

DATE\_FORMAT("012171", MMDDYY, MMDDYYYY) gibt die Zeichenfolge "01211971" zurück.

**Anmerkung:** Weitere Informationen zu gültigen Datumsformaten finden Sie unter „DATE“ auf Seite 52.

Zugehörige Funktionen

<b>Funktion</b>	<b>Beschreibung</b>
DATE	Konvertiert eine Datumszeichenfolge in ein julianisches Datum.
DATE_JULIAN	Gibt das julianische Datum der Eingabe zurück.
DATE_STRING	Gibt die Datumszeichenfolge des julianischen Datums zurück.

---

## DATE\_JULIAN-Makro

Das DATE\_JULIAN-Makro Makro ist nur in IBM Campaign verfügbar.

### Syntax

DATE\_JULIAN(year, month, day)

### Parameter

year

Gültige 2-stellige oder 4-stellige Jahreszahl.

month

Gültige Monatszahl zwischen 1 und 12.

day

Gültige Tageszahl zwischen 1 und 31.

### Beschreibung

DATE\_JULIAN gibt das julianische Datum der angegebenen Eingabe zurück. Das julianische Datum ist die Anzahl der seit dem 1. Januar 0000 vergangenen Tage.

### Beispiele

DATE\_JULIAN (2000,08,31) gibt die Zahl 730729 zurück.

### Zugehörige Funktionen

<b>Funktion</b>	<b>Beschreibung</b>
DATE	Konvertiert eine Datumszeichenfolge in ein julianisches Datum.
DATE_FORMAT	Konvertiert Datumsangaben von einem Format in ein anderes.
DATE_STRING	Gibt die Datumszeichenfolge des julianischen Datums zurück.

---

## DATE\_STRING-Makro

Das DATE\_STRING-Makro Makro ist nur in IBM Campaign verfügbar.

## Syntax

```
DATE_STRING(julian_date [, 'output_format'[, max_length]])  
DATE_STRING(julian_date [, 'format_string'[, max_length]])
```

## Parameter

julian\_date

Eine Zahl, die ein julianisches Datum darstellt, d. h. die Anzahl der seit dem 1. Januar 0000 vergangenen Tage.

output\_format

Zeichenfolge, gültiges Datumsformat.

max\_length

format\_string

Eine Formatierzeichenfolge, die optional eine beliebige Kombination der folgenden Formatcodes enthalten kann:

Code	Beschreibung
%a	Abgekürzter Wochentagsname
%A	Vollständiger Wochentagsname
%b	Abgekürzter Monatsname
%B	Vollständiger Monatsname
%c	Datums- und Uhrzeitdarstellung gemäß der Ländereinstellung
%d	Tag des Monats (01 - 31)
%H	Stunde im 24-Stunden-Format (00 - 23)
%I	Stunde im 12-Stunden-Format (01 - 12)
%j	Tag des Jahres (001 - 366)
%m	Monat (01 - 12)
%M	Minute (00 - 59)
%p	AM/PM-Anzeiger für 12-Stunden-Format gemäß der aktuellen Ländereinstellung
%S	Sekunde (00 - 59)
%U	Woche des Jahres, wobei Sonntag als erster Wochentag zählt (00 - 51)
%w	Wochentag (0 - 6; Sonntag ist 0)
%W	Woche des Jahres, wobei Montag als erster Wochentag zählt (00 - 51)
%x	Datumsdarstellung gemäß der aktuellen Ländereinstellung
%X	Uhrzeitdarstellung gemäß der aktuellen Ländereinstellung
%y	2-stelliges Jahr (00 - 99)

Code	Beschreibung
%Y	4-stelliges Jahr. Die voranstehenden Nullen im Jahr werden nicht abgeschnitten. Beispielsweise wird das Jahr 0201 als 0201 angezeigt und das Jahr 0001 wird als 0001 angezeigt.
%4Y	4-stelliges Jahr. Die voranstehenden Nullen im Jahr werden nicht abgeschnitten. Beispielsweise wird das Jahr 0201 als 0201 angezeigt und das Jahr 0001 wird als 0001 angezeigt.
%z, %Z	Name oder Abkürzung der Zeitzone; keine Ausgabe, falls Zeitzone unbekannt ist
%%	Prozentzeichen

## Beschreibung

DATE\_STRING gibt die Datumszeichenfolge des julianischen Datums zurück. Wenn output\_format nicht angegeben ist, wird das Standardschlüsselwort DELIM\_M\_D\_Y verwendet.

## Beispiele

DATE\_STRING(730729) gibt die Zeichenfolge "08/31/00" zurück.

**Anmerkung:** Weitere Informationen zu gültigen Datumsformaten finden Sie unter „DATE“ auf Seite 52.

## Zugehörige Funktionen

Funktion	Beschreibung
DATE	Konvertiert eine Datumszeichenfolge in ein julianisches Datum.
DATE_JULIAN	Gibt das julianische Datum der Eingabe zurück.
DATE_FORMAT	Konvertiert Datumsangaben von einem Format in ein anderes.

---

## DAY\_BETWEEN-Makro

Das DAY\_BETWEEN-Makro Makro ist nur in IBM Campaign verfügbar.

### Syntax

DAY\_BETWEEN(from\_date\_string, to\_date\_string [, input\_format])

### Parameter

from\_date\_string

Ein Text, der ein gültiges Datum darstellt, von dem aus die Anzahl der verstrichenen Tage berechnet werden soll.

to\_date\_string

Ein Text, der ein gültiges Datum darstellt, bis zu dem die Anzahl der Tage berechnet werden soll. Dieses Datum muss dasselbe Format wie from\_date\_string aufweisen.

input\_format

Eines der Schlüsselwörter in der folgenden Tabelle, das das Datumsformat von from\_date\_string und to\_date\_string angibt.

## Beschreibung

DAY\_BETWEEN gibt die Anzahl der Tage zwischen from\_date\_string und to\_date\_string zurück. Wenn input\_format nicht angegeben ist, wird das Standardschlüsselwort DELIM\_M\_D\_Y verwendet.

## Beispiele

DAY\_BETWEEN("08/25/00", "08/31/00") gibt die Zahl 6 zurück.

**Anmerkung:** Weitere Informationen zu gültigen Datumsformaten finden Sie unter „DATE“ auf Seite 52.

## Zugehörige Funktionen

Funktion	Beschreibung
DAY_FROMNOW	Gibt die Anzahl der Tage vom aktuellen Datum bis zu einem angegebenen Datum zurück.
DAY_INTERVAL	Gibt die Anzahl der Tage zwischen zwei angegebenen Terminen zurück.

---

## DAY\_FROMNOW-Makro

Das DAY\_FROMNOW-Makro Makro ist nur in IBM Campaign verfügbar.

### Syntax

DAY\_FROMNOW(to\_year, to\_month, to\_day)

### Parameter

to\_year

Gültige 2-stellige oder 4-stellige Jahreszahl.

to\_month

Gültige Monatszahl zwischen 1 und 12.

to\_day

Gültige Tageszahl zwischen 1 und 31.

## Beschreibung

DAY\_FROMNOW gibt die Anzahl der Tage zwischen dem aktuellen Tag und dem durch to\_year/to\_month/to\_day angegebenen Datum zurück.

**Anmerkung:** Wenn das angegebene Datum in der Vergangenheit liegt, ist der Rückgabewert negativ.

## Beispiele

Wenn das aktuelle Datum der 31. August 2000 ist, gibt DAY\_FROMNOW(2000,12,31) die Zahl 122 zurück.

## Zugehörige Funktionen

Funktion	Beschreibung
DAY_BETWEEN	Gibt die Anzahl der Tage zwischen zwei angegebenen Datumszeichenfolgen zurück.
DAY_INTERVAL	Gibt die Anzahl der Tage zwischen zwei angegebenen Terminen zurück.

---

## DAY\_INTERVAL-Makro

Das DAY\_INTERVAL-Makro Makro ist nur in IBM Campaign verfügbar.

### Syntax

DAY\_INTERVAL(from\_year, from\_month, from\_day, to\_year, to\_month, to\_day)

### Parameter

from\_year

Gültige 2-stellige oder 4-stellige Jahreszahl.

from\_month

Gültige Monatszahl zwischen 1 und 12.

from\_day

Gültige Tageszahl zwischen 1 und 31.

to\_year

Gültige 2-stellige oder 4-stellige Jahreszahl.

to\_month

Gültige Monatszahl zwischen 1 und 12.

to\_day

Gültige Tageszahl zwischen 1 und 31.

## Beschreibung

DAY\_INTERVAL gibt die Anzahl der Tage zwischen dem angegebenen Anfangsdatum (from\_year/from\_month/from\_day) und dem angegebenen Enddatum to\_year/to\_month/to\_day) zurück.

## Beispiele

DAY\_INTERVAL(2000,8,31,2000,12,31) gibt die Zahl 122 zurück.

## Zugehörige Funktionen

Funktion	Beschreibung
DAY_BETWEEN	Gibt die Anzahl der Tage zwischen zwei angegebenen Datumszeichenfolgen zurück.
DAY_FROMNOW	Gibt die Anzahl der Tage vom aktuellen Datum bis zu einem angegebenen Datum zurück.

---

## DAYOF-Makro

Das DAYOF-Makro Makro ist nur in IBM Campaign verfügbar.

### Syntax

DAYOF(date\_string [, input\_format])

### Parameter

date\_string

Ein Text, der ein gültiges Datum darstellt.

input\_format

Eines der Schlüsselwörter in der folgenden Tabelle, das das Datumsformat von date\_string angibt.

### Beschreibung

DAYOF gibt für das durch date\_string angegebene Datum den Tag des Monats als Zahl zurück. Wenn input\_format nicht angegeben ist, wird das Standardschlüsselwort DELIM\_M\_D\_Y verwendet.

### Beispiele

DAYOF("08/31/00") gibt die Zahl 31 zurück.

**Anmerkung:** Weitere Informationen zu gültigen Datumsformaten finden Sie unter „DATE“ auf Seite 52.

---

## DISTANCE-Makro

Das DISTANCE-Makro Makro ist in IBM Campaign und IBM Interact verfügbar.

## Syntax

```
DISTANCE(lat1, long1, lat2, long2[, UNIT_OF_MEASURE][, PRECISION])
```

## Parameter

lat1

Der Breitengrad des ersten Punkts als Dezimalwert.

long1

Der Längengrad des ersten Punkts als Dezimalwert.

lat2

Der Breitengrad des zweiten Punkts als Dezimalwert.

long2

Der Längengrad des zweiten Punkts als Dezimalwert.

UNIT\_OF\_MEASURE

Ein optionaler Parameter, der die Maßeinheit für den zurückgegebenen Abstand angibt. Mögliche Werte sind MILES (Meilen) oder KILOMETERS (Kilometer). Wenn Sie diesen Parameter nicht angeben, wird als Standardeinstellung MILES verwendet.

PRECISION

Ein optionaler Parameter, der die Genauigkeit nach dem Dezimalzeichen für den zurückgegebenen Abstand angibt. Wenn Sie einen Genauigkeitswert festlegen, wird der zurückgegebene Abstand auf die angegebene Anzahl von Dezimalstellen gekürzt. Der Maximalwert ist 5. Wenn Sie hierfür keinen Wert angeben, wird die Anzahl der Dezimalstellen nicht gekürzt.

## Beschreibung

DISTANCE berechnet den Abstand zwischen zwei Punkten. Breitengrad und Längengrad werden in Dezimalzahleinheiten erwartet. Verwenden Sie zum Trennen numerischer Werte immer ein Komma und ein Leerzeichen. Das ist notwendig, um Sprachen zu berücksichtigen, in denen ein Komma als Dezimaltrennzeichen verwendet wird, wie im zweiten Beispiel unten gezeigt wird.

## Beispiele

DISTANCE (18.529747, 73.839798, 18.533511, 73.8777995, MILES, 2) gibt den Wert von 2,50 Meilen zurück.

DISTANCE (18,529747, 73,839798, 18,533511, 73,8777995, KILOMETERS, 1) gibt den Wert von 4,0 Kilometern zurück.

---

## DIV-Makro

Das DIV-Makro Makro ist in IBM Campaign und IBM Interact verfügbar.

## Syntax

data DIV divisor data / divisor

## Parameter

data

Die numerischen Werte, die dividiert werden sollen. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

divisor

Der Wert oder die Werte, durch den oder die die Werte im angegebenen Datenbereich dividiert werden sollen. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Die Anzahl der Spalten in data2 muss mit der Anzahl der Spalten in data1 übereinstimmen, es sei denn, bei data2 handelt es sich um eine Konstante. Informationen zur Formatdefinition von divisor (identisch mit data) finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

## Beschreibung

DIV dividiert den angegebenen Datenbereich durch den Wert des Divisors. DIV gibt für jede Eingabespalte eine neue Spalte zurück, wobei jeweils die entsprechende Spalte von data durch die entsprechende Spalte von divisor dividiert wird (d. h., die erste Spalte von data wird durch die erste Spalte von divisor dividiert, die zweite Spalte durch die zweite Spalte usw.).

Wenn es sich bei divisor um eine Konstante handelt, wird jeder Wert in data durch den betreffenden Wert dividiert. Wenn data2 eine oder mehrere Spalten enthält, werden die Berechnungen zeilenweise zwischen einer Spalte von data2 und einer Spalte von data2 ausgeführt. Die erste Zeile von data wird durch die erste Zeile von divisor dividiert, die zweite Zeile durch die zweite Zeile usw. Diese zeilenweise ausgeführte Berechnung erzeugt für jede Zeile ein Ergebnis, bis zum letzten Wert der kürzesten Spalte.

**Anmerkung:** Wenn eine Spalte in jeder Zeile dieselbe Zahl  $x$  wie divisor enthält, ist dies dasselbe, als wenn als divisor die Konstante  $x$  verwendet wird.

**Anmerkung:** Der Operator DIV kann durch einen Schrägstrich (/) abgekürzt werden.

## Beispiele

TEMP = 8 DIV 4 oder TEMP = 8/4

Erstellt eine neue Spalte TEMP, die den Wert zwei enthält.

TEMP = V1/8

Erstellt eine neue Spalte TEMP, in der jeder Wert den Inhalt von Spalte V1 darstellt, der durch acht dividiert ist.

TEMP = V1:V3/2
Erstellt drei neue Spalten TEMP, VX und VY. Die Werte in der Spalte TEMP sind der durch zwei dividierte Inhalt von Spalte V1, die Werte von Spalte VX sind der durch zwei dividierte Inhalt von Spalte V2 und die Werte von Spalte VY sind der durch zwei dividierte Inhalt von Spalte V3.
TEMP = V1/V1
Erstellt eine neue Spalte TEMP, die nur Einsen enthält (jeder Wert, der durch sich selbst dividiert wird, ergibt eins).
TEMP = V1/V2
Erstellt eine neue Spalte TEMP, wobei jeder Zeilenwert von Spalte V1 durch den entsprechenden Zeilenwert von Spalte V2 dividiert ist.
TEMP = V1:V3/V4:V6
Erstellt drei neue Spalten TEMP, VX und VY. In der Spalte TEMP sind die Werte von Spalte V1 durch die entsprechenden Zeilenwerte von Spalte V4 dividiert. Die Spalte VX enthält die Division von Spalte V2 durch Spalte V5. Die Spalte VY enthält die Division von Spalte V3 durch Spalte V6.
TEMP = V1[10:20] / V2 oder TEMP = V1[10:20] / V2[1:11]
Erstellt eine neue Spalte TEMP, deren erste 11 Zellen das Ergebnis der Division der Werte in Zeile 10-20 von Spalte V1 durch die Werte in Zeile 1-11 von Spalte V2 enthalten. Die anderen Zellen in TEMP sind leer.

## Zugehörige Funktionen

Funktion	Beschreibung
MOD	Berechnet den Modulo-Wert des Inhalts des angegebenen Datenbereichs.
MULT	Multipliziert die Inhalte zweier Datenbereiche.
POW	Erhebt einen Basiswert in die Potenz des angegebenen Exponenten.

## EQ-Makro

Das EQ-Makro Makro ist in IBM Campaign und IBM Interact verfügbar.

### Syntax

data1 EQ data2 data1 == data2 (data1 = data2)

### Parameter

data1

Der Zellenbereich, der verglichen werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

data2

Die Zahlen, mit denen alle Werte in der angegebenen Spalte verglichen werden sollen. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Die Anzahl der Spalten in data2 muss mit der Anzahl der Spalten in data1 übereinstimmen, es sei denn, bei data2 handelt es sich um eine Konstante. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

## Beschreibung

EQ vergleicht die beiden angegebenen Datenbereiche und gibt eine Eins zurück, wenn die Werte gleich sind, bzw. eine Null, wenn die Werte ungleich sind. EQ gibt für jede Eingabespalte eine neue Spalte zurück, wobei jeweils die entsprechende Spalte von data1 mit der entsprechenden Spalte von data2 verglichen wird (d. h., die erste Spalte von data1 wird mit der ersten Spalte von data2 verglichen, die zweite Spalte mit der zweiten Spalte usw.).

Wenn es sich bei data2 um eine Konstante handelt, wird jeder Wert in data1 mit dem betreffenden Wert verglichen. Wenn es sich bei data2 um eine Spalte handelt, werden die Berechnungen zeilenweise ausgeführt. Die erste Zeile von data1 wird mit der ersten Zeile von data2 verglichen, die zweite Zeile mit der zweiten Zeile usw. Diese zeilenweise ausgeführte Berechnung erzeugt für jede Zeile ein Ergebnis, bis zum letzten Wert der kürzesten Spalte.

Beim Vergleich von Zeichenfolgen spielt die Groß-/Kleinschreibung keine Rolle (d. h., "Ja", "ja", "JA" und "jA" werden gleich behandelt).

**Anmerkung:** Der Operator EQ kann durch ein doppeltes Gleichheitszeichen (==) abgekürzt werden. Innerhalb runder Klammern kann auch ein einfaches Gleichheitszeichen (=) für die Makrofunktion EQ verwendet werden (außerhalb runder Klammern wird das Gleichheitszeichen als Zuweisungsoperator interpretiert).

## Beispiele

TEMP = 3 EQ 4 oder TEMP = 3==4 oder TEMP = (3=4)
Erstellt eine neue Spalte TEMP, die den Wert null enthält (drei ist ungleich vier).
TEMP = "Ja" == "JA"
Erstellt eine neue Spalte TEMP, die den Wert eins enthält (beim Vergleich von Zeichenfolgen wird die Groß-/Kleinschreibung nicht beachtet).
TEMP = V1 == 8
Erstellt eine neue Spalte TEMP, in der jeder Wert eins ist, wenn der entsprechende Zeilenwert von Spalte V1 gleich der Zahl acht ist; andernfalls ist der Wert null.
TEMP = V1==V1
Erstellt eine neue Spalte TEMP, die nur Einsen enthält (jede Zahl ist gleich sich selbst).
TEMP = V1==V2
Erstellt eine neue Spalte TEMP, wobei jeder Zeilenwert von Spalte V1 mit dem entsprechenden Zeilenwert von Spalte V2 verglichen wird.

TEMP = V1:V3 == V4:V6
Erstellt drei neue Spalten TEMP, VX und VY. In der Spalte TEMP sind die Werte von Spalte V1 mit den entsprechenden Zeilenwerten von Spalte V4 verglichen. Die Spalte VX enthält den Vergleich von Spalte V2 mit Spalte V5. Die Spalte VY enthält den Vergleich von Spalte V3 mit Spalte V6.
TEMP = V1[10:20] == V2 oder TEMP = V1[10:20] == V2[1:11]
Erstellt eine neue Spalte TEMP, deren erste 11 Zellen das Ergebnis des Vergleichs der Werte in Zeile 10-20 von Spalte V1 mit den Werten in Zeile 1-11 von Spalte V2 enthalten. Die anderen Zellen in TEMP sind leer.

## Zugehörige Funktionen

Funktion	Beschreibung
EQ	Gibt TRUE zurück, wenn der eine Datenbereich dem anderen gleich ist.
GE	Gibt TRUE zurück, wenn der eine Datenbereich größer-gleich dem anderen ist.
GT	Gibt TRUE zurück, wenn der eine Datenbereich größer als der andere ist.
LE	Gibt TRUE zurück, wenn der eine Datenbereich kleiner-gleich dem anderen ist.
LT	Gibt TRUE zurück, wenn der eine Datenbereich kleiner als der andere ist.
NE	Gibt TRUE zurück, wenn der eine Datenbereich dem anderen ungleich ist.

---

## EXP-Makro

Das EXP-Makro Makro ist nur in IBM Campaign verfügbar.

### Syntax

EXP(data)

### Parameter

data

Die numerischen Werte, die als Exponenten für die natürliche Zahl  $e$  verwendet werden. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

### Beschreibung

EXP erhebt die natürliche Zahl,  $e$ , in die Potenz, die jeweils durch den Wert im angegebenen Datenbereich angegeben wird (berechnet wird also  $e^x$ ). Die Konstante  $e$  hat den Wert 2,7182818. EXP gibt für jede Eingabespalte eine neue Spalte zurück,

die das Ergebnis  $e^x$  für jeden Wert  $x$  in der entsprechenden Eingabespalte enthält. EXP ist die Umkehrfunktion der Makrofunktion LN.

**Anmerkung:** Wenn der Wert  $x$  zu groß oder zu klein ist, wird ein Überlauferfehler zurückgegeben. Dies ist der Fall, wenn  $e^x$  den maximalen oder minimalen 32-Bit-Gleitkommawert überschreitet.

## Beispiele

TEMP = EXP(2)
Erstellt eine neue Spalte TEMP, die den Wert 7.39 enthält.
TEMP = EXP(V1)
Erstellt eine neue Spalte TEMP, in der jeweils $e$ in die Potenz erhoben wird, die durch den Inhalt von Spalte V1 angegeben wird.
TEMP = EXP(V1:V3)
Erstellt drei neue Spalten TEMP, VX und VY. Die Werte in der Spalte TEMP ergeben sich durch Potenzierung von $e$ um die in Spalte V1 angegebenen Werte, die Werte von Spalte VX ergeben sich durch Potenzierung von $e$ um die in Spalte V2 angegebenen Werte und die Werte von Spalte VY ergeben sich durch Potenzierung von $e$ um die in Spalte V3 angegebenen Werte.
TEMP = EXP(V1[10:20])
Erstellt eine neue Spalte TEMP, in der die ersten 11 Zellen jeweils die Potenzierung von $e$ um die Werte in Zeile 10-20 von Spalte V1 darstellen. Die anderen Zellen in TEMP sind leer.
TEMP = EXP(V1[1:5]:V2)
Erstellt zwei neue Spalten TEMP und VX, die jeweils die Werte in Zeile 1-5 enthalten (die anderen Zellen sind leer). Die Werte in der Spalte TEMP stellen die Potenzierung von $e$ um die entsprechenden Zeilenwerte von Spalte V1 dar, die Werte in der Spalte VX stellen die Potenzierung von $e$ um die entsprechenden Zeilenwerte von Spalte V2 dar.

## Zugehörige Funktionen

Funktion	Beschreibung
LN oder LOG	Berechnet den natürlichen Logarithmus des Inhalts des angegebenen Datenbereichs.
LOG2	Berechnet den binären Logarithmus des Inhalts des angegebenen Datenbereichs.
LOG10	Berechnet den dekadischen Logarithmus des Inhalts des angegebenen Datenbereichs.
POW	Erhebt einen Basiswert in die Potenz des angegebenen Exponenten.

---

## EXTERNALCALLOUT-Makro

Das EXTERNALCALLOUT-Makro ist nur in IBM Interact verfügbar.

### Syntax

EXTERNALCALLOUT( *calloutName*, *Arg1*, ...)

## Parameter

calloutName

Der Name des mithilfe der ExternalCallout-API erstellten Aufrufs. Dieser Name muss mit dem Namen der ExternalCallout-Kategorie übereinstimmen, die Sie in IBM Marketing Platform erstellt haben.

Arg1

Ein für Ihren Aufruf gegebenenfalls erforderliches Argument.

## Beschreibung

Mit EXTERNALCALLOUT können Sie eine externe Anwendung aufrufen, um Ihrem interaktiven Ablaufdiagramm Daten hinzuzufügen. EXTERNALCALLOUT kann alle Daten zurückgeben, die von Ihrem Aufruf zurückgegeben werden. Sie müssen diesen Aufruf mithilfe der ExternalCallout-API in Java schreiben. Weitere Informationen finden Sie im *IBM Interact-Administratorhandbuch*.

## Beispiele

```
EXTERNALCALLOUT(getStockPrice, UNCA)
```

Ruft den Aufruf `getStockPrice` auf und übergibt den Namen der Aktie, `UNCA`, als Argument. Dieser benutzerdefinierte Aufruf gibt den Aktienkurs zurück, wie er durch den Aufruf definiert ist.

---

## FACTORIAL-Makro

Das FACTORIAL-Makro Makro ist nur in IBM Campaign verfügbar.

### Syntax

```
FACTORIAL(data)
```

### Parameter

data

Die Ganzzahlwerte, deren Fakultät berechnet werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt; der Wert muss jedoch größer-gleich null sein. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

### Beschreibung

FACTORIAL berechnet die Fakultät der Werte im angegebenen Datenbereich. Alle Eingabewerte müssen Ganzzahlen größer-gleich null sein. Die Fakultät einer Ganzzahl kleiner-gleich eins ist eins. Bei Ganzzahlen  $x \geq 2$  ist die Fakultät  $x! = x(x-1)(x-2)\dots(x-(x-1))$ . FACTORIAL gibt für jede Eingabespalte eine neue Spalte zurück, die die Fakultät der Zahlen in der entsprechenden Eingabespalte enthält.

**Anmerkung:** Jeder Wert größer als 34 erzeugt die Ausgabe ??? (Gleitkomma-Überlauffehler).

## Beispiele

TEMP = FACTORIAL(3)
Erstellt eine neue Spalte TEMP, die den Wert 6 enthält.
TEMP = FACTORIAL(-2)
Generiert einen Fehler 333, der darauf hinweist, dass das Argument größer-gleich 0 sein muss.
TEMP = FACTORIAL(V1)
Erstellt eine neue Spalte TEMP, in der jeder Wert die Fakultät des Inhalts von Spalte V1 darstellt.
TEMP = FACTORIAL(V1:V3)
Erstellt drei neue Spalten TEMP, VX und VY. Die Werte in der Spalte TEMP sind jeweils die Fakultät des Inhalts von Spalte V1, die Werte von Spalte VX sind die Fakultät des Inhalts von Spalte V2 und die Werte von Spalte VY sind die Fakultät des Inhalts von Spalte V3.
TEMP = FACTORIAL(V1[10:20])
Erstellt eine neue Spalte TEMP, in der die ersten 11 Zellen jeweils die Fakultät der Werte in Zeile 10-20 von Spalte V1 darstellen. Die anderen Zellen in TEMP sind leer.
TEMP = FACTORIAL(V1[50:99]:V2)
Erstellt zwei neue Spalten TEMP und VX, die jeweils Werte in Zeile 1-50 enthalten (die anderen Zellen sind leer). Die Werte in der Spalte TEMP sind jeweils die Fakultät der Zeilen von Spalte V1, die Werte in der Spalte VX sind die Fakultät der Zeilen von Spalte V2.

---

## FLOOR-Makro

Das FLOOR-Makro Makro ist nur in IBM Campaign verfügbar.

### Syntax

FLOOR(data)

### Parameter

data

Die numerischen Werte, deren Untergrenze berechnet werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

### Beschreibung

FLOOR berechnet die Untergrenze der Werte im angegebenen Datenbereich. Die Untergrenze einer Zahl ist die größte Ganzzahl, die kleiner als die Zahl ist. FLOOR gibt für jede Eingabespalte eine neue Spalte zurück, die die Untergrenze der Zahlen in der entsprechenden Eingabespalte enthält.

**Anmerkung:** FLOOR ist mit der Makrofunktion INT identisch.

## Beispiele

TEMP = FLOOR(4.3)
Erstellt eine neue Spalte TEMP, die den Wert 4 enthält.
TEMP = FLOOR(-2.9)
Erstellt eine neue Spalte TEMP, die den Wert -3 enthält.
TEMP = FLOOR(V1)
Erstellt eine neue Spalte TEMP, in der jeder Wert die Untergrenze des Inhalts von Spalte V1 darstellt.
TEMP = FLOOR(V1:V3)
Erstellt drei neue Spalten TEMP, VX und VY. Die Werte in der Spalte TEMP sind jeweils die Untergrenze des Inhalts von Spalte V1, die Werte von Spalte VX sind die Untergrenze des Inhalts von Spalte V2 und die Werte von Spalte VY sind die Untergrenze des Inhalts von Spalte V3.
TEMP = FLOOR(V1[10:20])
Erstellt eine neue Spalte TEMP, in der die ersten 11 Zellen jeweils die Untergrenze der Werte in Zeile 10-20 von Spalte V1 darstellen. Die anderen Zellen in TEMP sind leer.
TEMP = FLOOR(V1[50:99]:V2)
Erstellt zwei neue Spalten TEMP und VX, die jeweils Werte in Zeile 1-50 enthalten (die anderen Zellen sind leer). Die Werte in der Spalte TEMP sind jeweils die Untergrenze der Zeilen von Spalte V1, die Werte in der Spalte VX sind die Untergrenze der Zeilen von Spalte V2.

## Zugehörige Funktionen

Funktion	Beschreibung
CEILING	Berechnet die Obergrenze jedes Werts im angegebenen Datenbereich.
FRACTION	Berechnet die Nachkommastellen jedes Werts im angegebenen Datenbereich.
TRUNCATE	Berechnet den Ganzzahlanteil jedes Werts im angegebenen Datenbereich.

---

## FORMAT-Makro

Das FORMAT-Makro Makro ist nur in IBM Campaign verfügbar.

### Syntax

Format weist zwei Formen auf, eine für numerische Datentypen und eine für Text-/Zeichendatentypen.

Bei numerischen Datentypen:

```
FORMAT(colName, width [, precision [, format_type [, alignment [, padding]]]])
```

Bei Text-/Zeichendatentypen:

FORMAT(colName, width [, alignment])

## Parameter

colName

Das Makro untersucht colName und ermittelt dessen Datentyp, um dementsprechend die jeweiligen Regeln für nachfolgende Parameter anzuwenden.

width

Die Breite muss groß genug sein, um das vollständige Ergebnis aufnehmen zu können; andernfalls wird das Ergebnis abgeschnitten. Die zulässigen Werte reichen von 1 bis 29, wenn colName numerisch ist, andernfalls von 1 bis 255.

precision

Die Genauigkeit ist die Anzahl der Stellen nach dem Dezimalzeichen. Die zulässigen Werte reichen von 0 bis 15. Wenn sie null ist, ist das Ergebnis eine Ganzzahl. Der Standardwert für die Genauigkeit ist 2.

format\_type

Gültige Schlüsselwörter für format\_type:

PERIOD	Ein Punkt (.) wird als Dezimalzeichen verwendet. Es wird kein Trennzeichen verwendet. Dies ist der Standardwert.
COMMA	Als Dezimalzeichen wird ein Komma (,) verwendet. Es wird kein Trennzeichen verwendet.
PERIOD_COMMA	Als Dezimalzeichen wird ein Punkt und als Trennzeichen ein Komma verwendet.
COMMA_PERIOD	Als Dezimalzeichen wird ein Komma und als Trennzeichen ein Punkt verwendet.

alignment

Gültige Schlüsselwörter für alignment sind LEFT und RIGHT. Bei numerischen Datentypen ist der Standardwert RIGHT, bei Text-/Zeichendatentypen LEFT.

padding

Gültige Schlüsselwörter für padding sind SPACE und ZERO. Der Standardwert ist SPACE. Wenn alignment den Wert LEFT aufweist, wird ZERO ignoriert (und stattdessen SPACE verwendet).

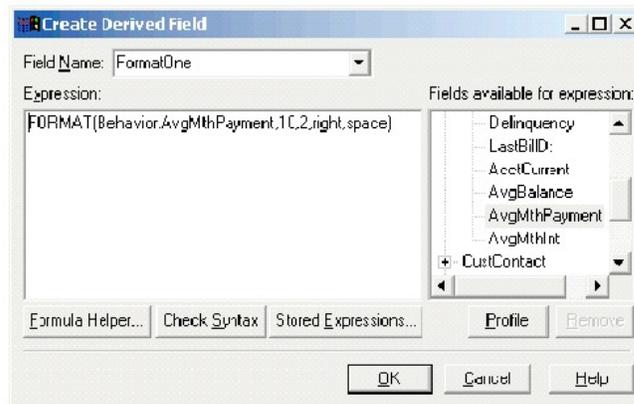
Numerische Zeichenfolgen innerhalb eines Text-/Zeichendatentyps werden als Text/Zeichen behandelt. Außerdem unterstützt die numerische Form mehrere optionale Schlüsselwörter, die jeweils einen Standardwert aufweisen. Um jedoch den Standardwert des zweiten oder eines nachfolgenden optionalen Schlüsselworts zu überschreiben, MÜSSEN Sie die Standardwerte für die vorhergehenden optionalen Schlüsselwörter angeben (sie werden somit obligatorisch). Um zum Beispiel den Standardwert für alignment durch LEFT zu überschreiben, müssen Sie Folgendes codieren: FORMAT(myNumCol, 10, 2, PERIOD, LEFT).

## Beschreibung

FORMAT konvertiert numerische Daten in eine Zeichenfolgeform, wobei die Ausgabezeichenfolge durch verschiedene Formatierungsoptionen gesteuert und definiert werden kann. Dies ist besonders nützlich, wenn Momentaufnahmedateien mit bestimmten Formaten für das Mailing von Dateien erstellt werden sollen.

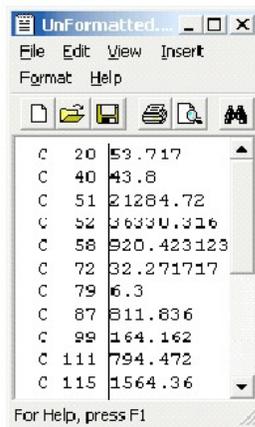
## Beispiele

Das folgende Beispiel definiert mithilfe von FORMAT ein abgeleitetes Feld.



Die folgenden Beispiele zeigen dasselbe Feld, AvgMthPayment, in drei Formaten.

Unformatiert:



Mit `FORMAT(Behavior.AvgMthPayment,10,2,right.space)` formatiert:

Label	Value
C 20	53.72
C 40	43.80
C 51	21284.72
C 52	36330.32
C 58	920.42
C 72	32.27
C 79	6.30
C 87	811.84
C 99	164.16
C 111	794.47
C 115	1564.36

Mit `FORMAT(Behavior.AvgMthPayment,10,4)` formatiert:

Label	Value
C 20	53.7170
C 40	43.8000
C 51	21284.7200
C 52	36330.3160
C 58	920.4231
C 72	32.2717
C 79	6.3000
C 87	811.8360
C 99	164.1620
C 111	794.4720
C 115	1564.3600

---

## FRACTION-Makro

Das FRACTION-Makro Makro ist nur in IBM Campaign verfügbar.

### Syntax

`FRACTION(data)`

### Parameter

data

Die numerischen Werte, deren Nachkommastellen berechnet werden sollen. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

## Beschreibung

FRACTION berechnet die Nachkommastellen der Werte im angegebenen Datenbereich. Dabei wird für jede Eingabespalte eine neue Spalte zurückgegeben, die die Nachkommastellen der Zahlen in der entsprechenden Eingabespalte enthält.

**Anmerkung:** Die Makrofunktionen FRACTION und TRUNCATE sind komplementär, da sie in der Summe die ursprünglichen Werte ergeben.

## Beispiele

TEMP = FRACTION(4.3)
Erstellt eine neue Spalte TEMP, die den Wert 0.3 enthält.
TEMP = FRACTION(2.9)
Erstellt eine neue Spalte TEMP, die den Wert -0.9 enthält.
TEMP = FRACTION(V1)
Erstellt eine neue Spalte TEMP, in der jeder Wert die Nachkommastellen des Inhalts von Spalte V1 darstellt.
TEMP = FRACTION(V1:V3)
Erstellt drei neue Spalten TEMP, VX und VY. Die Werte in der Spalte TEMP sind jeweils die Nachkommastellen des Inhalts von Spalte V1, die Werte von Spalte VX sind die Nachkommastellen des Inhalts von Spalte V2 und die Werte von Spalte VY sind die Nachkommastellen des Inhalts von Spalte V3.
TEMP = FRACTION(V1[10:20])
Erstellt eine neue Spalte TEMP, in der die ersten 11 Zellen jeweils die Nachkommastellen der Werte in Zeile 10-20 von Spalte V1 enthalten. Die anderen Zellen in TEMP sind leer.
TEMP = FRACTION(V1[50:99]:V2)
Erstellt zwei neue Spalten TEMP und VX, die jeweils Werte in Zeile 1-50 enthalten (die anderen Zellen sind leer). Die Werte in der Spalte TEMP sind jeweils die Nachkommastellen der Zeilen von Spalte V1, die Werte in der Spalte VX sind die Nachkommastellen der Zeilen von Spalte V2.

## Zugehörige Funktionen

Funktion	Beschreibung
CEILING	Berechnet die Obergrenze jedes Werts im angegebenen Datenbereich.
FLOOR	Berechnet die Untergrenze jedes Werts im angegebenen Datenbereich.
TRUNCATE	Berechnet den Ganzzahlanteil jedes Werts im angegebenen Datenbereich.

---

## GE-Makro

Das GE-Makro ist in IBM Campaign und IBM Interact verfügbar.

## Syntax

```
data1 GE data2 data1 >= data2
```

## Parameter

data1

Der numerische Zellenbereich, der verglichen werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

data2

Die Zahlen, mit denen alle Werte in der angegebenen Spalte verglichen werden sollen. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Die Anzahl der Spalten in data2 muss mit der Anzahl der Spalten in data1 übereinstimmen, es sei denn, bei data2 handelt es sich um eine Konstante. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

## Beschreibung

GE vergleicht die beiden angegebenen Datenbereiche und gibt eine Eins zurück, wenn die Werte im ersten Dataset größer-gleich den Werten im zweiten Dataset sind, andernfalls eine Null. LE gibt für jede Eingabespalte eine neue Spalte zurück, wobei jeweils die entsprechende Spalte von data1 mit der entsprechenden Spalte von data2 verglichen wird (d. h., die erste Spalte von data1 wird mit der ersten Spalte von data2 verglichen, die zweite Spalte mit der zweiten Spalte usw.).

Wenn es sich bei data2 um eine Konstante handelt, wird jeder Wert in data1 mit dem betreffenden Wert verglichen. Wenn es sich bei data2 um eine Spalte handelt, werden die Berechnungen zeilenweise ausgeführt. Die erste Zeile von data1 wird mit der ersten Zeile von data2 verglichen, die zweite Zeile mit der zweiten Zeile usw. Diese zeilenweise ausgeführte Berechnung erzeugt für jede Zeile ein Ergebnis, bis zum letzten Wert der kürzesten Spalte.

**Anmerkung:** Der Operator GE kann durch ein Größer-als-Zeichen, auf das ein Gleichheitszeichen folgt, abgekürzt werden (>=).

## Beispiele

```
TEMP = 9 GE 4 oder TEMP = 9 >= 4
```

Erstellt eine neue Spalte TEMP, die den Wert eins enthält (neun ist größer als vier).

```
TEMP = V1 >= 8
```

Erstellt eine neue Spalte TEMP, in der jeder Wert eins ist, wenn der entsprechende Zeilenwert von Spalte V1 größer-gleich der Zahl acht ist; andernfalls ist der Wert null.

TEMP = V1:V3 >= 2
Erstellt drei neue Spalten TEMP, VX und VY. Die Werte in der Spalte TEMP sind der mit zwei verglichene Inhalt von Spalte V1, die Werte von Spalte VX sind der mit zwei verglichene Inhalt von Spalte V2 und die Werte von Spalte VY sind der mit zwei verglichene Inhalt von Spalte V3.
TEMP = V1 >= V1
Erstellt eine neue Spalte TEMP, die nur Einsen enthält (jede Zahl ist gleich sich selbst).
TEMP = V1 >= V2
Erstellt eine neue Spalte TEMP, wobei jeder Zeilenwert von Spalte V1 mit dem entsprechenden Zeilenwert von Spalte V2 verglichen wird.
TEMP = V1:V3 >= V4:V6
Erstellt drei neue Spalten TEMP, VX und VY. In der Spalte TEMP sind die Werte von Spalte V1 mit den entsprechenden Zeilenwerten von Spalte V4 verglichen. Die Spalte VX enthält den Vergleich von Spalte V2 mit Spalte V5. Die Spalte VY enthält den Vergleich von Spalte V3 mit Spalte V6.
TEMP = V1[10:20] >= V2 oder TEMP = V1[10:20] >= V2[1:11]
Erstellt eine neue Spalte TEMP, deren erste 11 Zellen das Ergebnis des Vergleichs der Werte in Zeile 10-20 von Spalte V1 mit den Werten in Zeile 1-11 von Spalte V2 enthalten. Die anderen Zellen in TEMP sind leer.

## Zugehörige Funktionen

NE	Gibt TRUE zurück, wenn der eine Datenbereich dem anderen ungleich ist.
----	--

---

## GROUPBY-Makro

Das GROUPBY-Makro Makro ist nur in IBM Campaign verfügbar.

### Syntax

GROUPBY(group\_field, keyword, rolled\_field [,output\_field])

### Parameter

- group\_field  
Gibt die Variable an, nach der die Einträge gruppiert werden (d. h., alle identischen Werte der angegebenen Variable werden derselben Gruppe zugeordnet).
- keyword  
Gibt die zusammenfassende Rollup-Funktion an, die auf das Rollup-Feld angewendet werden soll.
- rolled\_field  
Gibt die Variable an, die zusammengefasst werden soll.
- output\_field  
Gibt eine alternative Variable an, die für eine einzelne Zeile einer Gruppe zurückgegeben werden soll, und kann nur mit den Schlüsselwörtern MinOf, MaxOf und MedianOf eingesetzt werden.

## Beschreibung

GROUPBY fasst mehrere Zeilen von Daten in einer Gruppe zusammen. Die Ausgabe dieser Funktion ist eine einzelne Spalte. Die Ausgabe ist das Ergebnis der durch keyword angegebenen Operation bezüglich rolled\_field in der durch group\_field angegebenen homogenen Gruppe. Wenn es mehrere Antworten gibt, die eine bestimmte Bedingung erfüllen, wird die erste gefundene Antwort zurückgegeben.

Wenn das optionale output\_field nicht angegeben ist, ist die Ausgabe das Ergebnis der auf rolled\_field angewendeten Operation. Wenn output\_field angegeben ist, ist das Ergebnis das output\_field der Zeile in der Gruppe.

Wenn mehrere Zeilen in einer Gruppe die angegebene Bedingung erfüllen (etwa wenn bezüglich des Maximalwerts ein Gleichstand vorliegt), wird das output\_field zurückgegeben, das der ersten Zeile zugeordnet ist, die die Bedingung erfüllt.

**Anmerkung:** Wenn Sie eine mehrspaltige Gruppierung bearbeiten möchten, können Sie eine Liste von Feldnamen, die durch Kommas getrennt sind, in geschweifte Klammern "{ }" einschließen und im Aufruf des Makros GROUPBY als ersten Parameter verwenden.

Die folgenden Schlüsselwörter werden unterstützt (die Groß-/Kleinschreibung wird nicht beachtet):

Keyword	Zeichenfolge?	Beschreibung
	Ja/Nein	
CountOf	Ja	Gibt die Anzahl der Einträge in jeder Gruppe zurück (rolled_field kann numerisch oder eine Zeichenfolge sein; der Rückgabewert ist unabhängig vom Wert von rolled_field derselbe).
MinOf	Ja	Gibt den Minimalwert von rolled_field in jeder Gruppe zurück (rolled_field kann numerisch oder eine Zeichenfolge sein; wenn rolled_field eine Zeichenfolge ist, wird der Wert zurückgegeben, der bei alphabetischer Sortierung dem Anfang des Alphabets am nächsten liegt).
MaxOf	Ja	Gibt den Maximalwert von rolled_field in jeder Gruppe zurück (rolled_field kann numerisch oder eine Zeichenfolge sein; wenn rolled_field eine Zeichenfolge ist, wird der Wert zurückgegeben, der bei alphabetischer Sortierung dem Ende des Alphabets am nächsten liegt).

Keyword	Zeichenfolge?	Beschreibung
	Ja/Nein	
DiffOf	Ja	Gibt die Anzahl unterschiedlicher Werte von rolled_field in jeder Gruppe zurück (rolled_field kann numerisch oder eine Zeichenfolge sein).
AvgOf	Nein	Gibt den Durchschnittswert von rolled_field in jeder Gruppe zurück (rolled_field muss numerisch sein).
ModeOf	Ja	Gibt den Modalwert (d. h. den am häufigsten vorkommenden Wert) von rolled_field in jeder Gruppe zurück (rolled_field kann numerisch oder eine Zeichenfolge sein).
MedianOf	Ja	Gibt den gemittelten Wert (d. h. den Mittelwert bei Sortierung nach rolled_field) von rolled_field in jeder Gruppe zurück (rolled_field kann numerisch oder eine Zeichenfolge sein; wenn rolled_field eine Zeichenfolge ist, werden die Werte alphabetisch sortiert).
OrderOf	Ja	Gibt die Reihenfolge von rolled_field in jeder Gruppe zurück (rolled_field muss numerisch sein). Wenn mehrere Einträge denselben Wert haben, erhalten alle denselben Wert.
SumOf	Nein	Gibt die Summe von rolled_field in jeder Gruppe zurück (rolled_field muss numerisch sein).
StdevOf	Nein	Gibt die Standardabweichung von rolled_field in jeder Gruppe zurück (rolled_field muss numerisch sein).

Keyword	Zeichenfolge?	Beschreibung
	Ja/Nein	
IndexOf	Ja	Gibt den 1-basierten Index (nach rolled_field sortiert) jedes Datensatzes zurück (rolled_field kann numerisch oder eine Zeichenfolge sein). Die Sortierreihenfolge ist aufsteigend.  Hinweis: Bei numerischen Feldern kann die Sortierreihenfolge von RankOf und IndexOf absteigend gemacht werden, indem dem Sortierfeld ein Minuszeichen (-) vorangestellt wird.
RankOf	Ja	Gibt die 1-basierte Kategorie (nach rolled_field sortiert) zurück, in der jeder Datensatz liegt (rolled_field kann numerisch oder eine Zeichenfolge sein). Die Sortierreihenfolge ist aufsteigend.  Hinweis: Bei numerischen Feldern kann die Sortierreihenfolge von RankOf und IndexOf absteigend gemacht werden, indem dem Sortierfeld ein Minuszeichen (-) vorangestellt wird.

## Beispiele

```
GROUPBY (Household_ID, SumOf, Account_Balance)
```

Berechnet die Summe aller Kontostände für den jeweiligen Haushalt.

```
GROUPBY (Cust_ID, MinOf, Date(Account_Open_Date), Acc_Num)
```

Gibt die Kontonummer des ersten Kontos zurück, das vom Kunden eröffnet wurde.

---

## GROUPBY\_WHERE-Makro

Das GROUPBY\_WHERE-Makro Makro ist nur in IBM Campaign verfügbar.

### Syntax

```
GROUPBY_WHERE(group_field, keyword, rolled_field, where_value  
[,output_field])
```

## Parameter

- `group_field`  
Gibt die Variable an, nach der die Einträge gruppiert werden (d. h., alle identischen Werte der angegebenen Variable werden derselben Gruppe zugeordnet).
- `keyword`  
Gibt die zusammenfassende Rollup-Funktion an, die auf das Rollup-Feld angewendet werden soll.
- `rolled_field`  
Gibt die Variable an, die zusammengefasst werden soll.
- `where_value`  
Ein Ausdruck, dessen Auswertung den Wert eins oder null ergibt und der angibt, welche Zeilen in die Rollup-Operation einbezogen werden sollen.
- `output_field`  
Gibt eine alternative Variable an, die für eine einzelne Zeile einer Gruppe zurückgegeben werden soll, und kann nur mit den Schlüsselwörtern `MinOf`, `MaxOf` und `MedianOf` eingesetzt werden.

## Beschreibung

`GROUPBY_WHERE` fasst bestimmte Zeilen von Daten in einer Gruppe zusammen. Die Ausgabe dieser Funktion ist eine einzelne Spalte. Die Ausgabe ist das Ergebnis der durch `keyword` angegebenen Operation bezüglich `rolled_field` in der durch `group_field` angegebenen homogenen Gruppe, gefiltert durch `where_value`. Es werden nur Zeilen mit dem `where_value` eins in die Berechnung einbezogen.

Wenn das optionale `output_field` nicht angegeben ist, ist die Ausgabe das Ergebnis der auf `rolled_field` angewendeten Operation. Wenn `output_field` angegeben ist, ist das Ergebnis das `output_field` der Zeile in der Gruppe.

**Anmerkung:** Unter „GROUPBY-Makro“ auf Seite 76 finden Sie weitere Informationen zu gültigen Werten für `keyword`.

## Beispiele

```
GROUPBY_WHERE (Household_ID, SumOf, Account_Balance, Account_Balance>0)
```

Berechnet die Summe aller Konten mit positivem Saldo für den jeweiligen Haushalt.

```
GROUPBY_WHERE (Cust_ID, AvgOf, Purchase_Amt, Date(Current_Date) -  
Date(Purchase_Date)<90)
```

Berechnet den durchschnittlichen Kaufbetrag des jeweiligen Kunden bei Einkäufen in den vergangenen 90 Tagen.

---

## GT-Makro

Das GT-Makro Makro ist in IBM Campaign und IBM Interact verfügbar.

### Syntax

```
data1 GT data2 data1 > data2
```

## Parameter

data1

Der numerische Zellenbereich, der verglichen werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

data2

Die Zahlen, mit denen alle Werte in der angegebenen Spalte verglichen werden sollen. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Die Anzahl der Spalten in data2 muss mit der Anzahl der Spalten in data1 übereinstimmen, es sei denn, bei data2 handelt es sich um eine Konstante. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

## Beschreibung

GT vergleicht die beiden angegebenen Datenbereiche und gibt eine Eins zurück, wenn die Werte im ersten Dataset größer als die Werte im zweiten Dataset sind, andernfalls eine Null. GT gibt für jede Eingabespalte eine neue Spalte zurück, wobei jeweils die entsprechende Spalte von data1 mit der entsprechenden Spalte von data2 verglichen wird (d. h., die erste Spalte von data1 wird mit der ersten Spalte von data2 verglichen, die zweite Spalte mit der zweiten Spalte usw.).

Wenn es sich bei data2 um eine Konstante handelt, wird jeder Wert in data1 mit dem betreffenden Wert verglichen. Wenn es sich bei data2 um eine Spalte handelt, werden die Berechnungen zeilenweise ausgeführt. Die erste Zeile von data1 wird mit der ersten Zeile von data2 verglichen, die zweite Zeile mit der zweiten Zeile usw. Diese zeilenweise ausgeführte Berechnung erzeugt für jede Zeile ein Ergebnis, bis zum letzten Wert der kürzesten Spalte.

**Anmerkung:** Der Operator GT kann durch ein Größer-als-Zeichen abgekürzt werden (>).

## Beispiele

TEMP = 3 GT 4 oder TEMP = 3 > 4

Erstellt eine neue Spalte TEMP, die den Wert null enthält (drei ist nicht größer als vier).

TEMP = V1 > 8

Erstellt eine neue Spalte TEMP, in der jeder Wert eins ist, wenn der entsprechende Zeilenwert von Spalte V1 größer als die Zahl acht ist; andernfalls ist der Wert null.

TEMP = V1:V3 > 2

Erstellt drei neue Spalten TEMP, VX und VY. Die Werte in der Spalte TEMP sind der mit zwei verglichene Inhalt von Spalte V1, die Werte von Spalte VX sind der mit zwei verglichene Inhalt von Spalte V2 und die Werte von Spalte VY sind der mit zwei verglichene Inhalt von Spalte V3.

TEMP = V1 > V1
Erstellt eine neue Spalte TEMP, die nur Nullen enthält (keine Zahl ist größer als sie selbst).
TEMP = V1 > V2
Erstellt eine neue Spalte TEMP, wobei jeder Zeilenwert von Spalte V1 mit dem entsprechenden Zeilenwert von Spalte V2 verglichen wird.
TEMP = V1:V3 > V4:V6
Erstellt drei neue Spalten TEMP, VX und VY. In der Spalte TEMP sind die Werte von Spalte V1 mit den entsprechenden Zeilenwerten von Spalte V4 verglichen. Die Spalte VX enthält den Vergleich von Spalte V2 mit Spalte V5. Die Spalte VY enthält den Vergleich von Spalte V3 mit Spalte V6.
TEMP = V1[10:20] > V2 oder TEMP = V1[10:20] > V2[1:11]
Erstellt eine neue Spalte TEMP, deren erste 11 Zellen das Ergebnis des Vergleichs der Werte in Zeile 10-20 von Spalte V1 mit den Werten in Zeile 1-11 von Spalte V2 enthalten. Die anderen Zellen in TEMP sind leer.

## Zugehörige Funktionen

Funktion	Beschreibung
EQ	Gibt TRUE zurück, wenn der eine Datenbereich dem anderen gleich ist.
GE	Gibt TRUE zurück, wenn der eine Datenbereich größer-gleich dem anderen ist.
LE	Gibt TRUE zurück, wenn der eine Datenbereich kleiner-gleich dem anderen ist.
LT	Gibt TRUE zurück, wenn der eine Datenbereich kleiner als der andere ist.
NE	Gibt TRUE zurück, wenn der eine Datenbereich dem anderen ungleich ist.

---

## IF-Makro

Das IF-Makro Makro ist in IBM Campaign und IBM Interact verfügbar.

### Syntax

IF(predicate\_col, then\_value) IF(predicate\_col, then\_value, else\_value)

### Parameter

predicate\_col

Eine Spalte mit booleschen Werten oder ein Ausdruck, dessen Auswertung eine einzelne Spalte mit booleschen Werten ergibt. Boolesche Werte werden als null oder ungleich null interpretiert. Diese Spalte sollte mindestens so viele Zeilen enthalten wie der Datenbereich, aus dem die data extrahiert werden.

then\_value

Der oder die Werte, die zurückgegeben werden, wenn die entsprechende Zeile von predicate\_col einen Wert ungleich null enthält. Dabei kann es sich um einen kons-

tanten Wert, eine Spalte oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Unter „Makrofunktionsparameter für IBM Campaign“ auf Seite 9 finden Sie weitere Informationen zur Formatdefinition von `then_value` (identisch mit `data`).

`else_value`

Wenn dieser optionale Parameter angegeben wird, wird dieser Wert zurückgegeben, falls die entsprechende Zeile von `predicate_col` eine Null enthält. Dabei kann es sich um einen konstanten Wert, eine Spalte oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Wenn `else_value` nicht angegeben wird, wird jedes Mal eine Null zurückgegeben, wenn die Auswertung von `predicate_col` 'falsch' ergibt. Unter „Makrofunktionsparameter für IBM Campaign“ auf Seite 9 finden Sie weitere Informationen zur Formatdefinition von `else_value` (identisch mit `data`).

## Beschreibung

IF wertet den Ausdruck in `predicate_col` aus und gibt `then_value` zurück, wenn der Ausdruck wahr ist, bzw. `else_value`, wenn der Ausdruck falsch ist. In `then_value` und in `else_value` wird dieselbe Anzahl von Spalten zurückgegeben. Die neuen Spalten enthalten die entsprechenden Zeilenwerte aus `then_value`, wenn der Wert von `predicate_col` ungleich null ist. Wenn `else_value` angegeben wird, wird dieser Wert zurückgegeben, falls der Wert von `predicate_col` null ist. Wenn `else_value` nicht angegeben wird, wird null zurückgegeben.

Da IF zeilenweise vorgeht, wird für jede Zeile bis zum letzten Wert der kürzesten Spalte (d. h. der kürzesten Spalte von `predicate_col`, `then_value` und `else_value`) ein Ergebnis erzeugt.

**Anmerkung:** In der Regel empfiehlt es sich, eine Prädikatspalte zu erstellen, die eine der Vergleichsmakrofunktionen verwendet (`==`, `>`, `<`, `ISEVEN`, `ISODD` usw.).

## Beispiele

<code>TEMP = IF(1, V1)</code>	Erstellt eine neue Spalte TEMP, die eine Kopie von Spalte V1 enthält.
<code>TEMP = IF(V1, 1, 0)</code>	Erstellt eine neue Spalte TEMP, in der jeder Wert eins ist, wenn der entsprechende Zeilenwert von Spalte V1 ungleich null ist; andernfalls ist der Wert null.
<code>TEMP = IF(V3, V1, V2)</code>	Erstellt eine neue Spalte TEMP, in der jeder Wert aus Spalte V1 kopiert wird, wenn der entsprechende Wert von Spalte V3 ungleich null ist; andernfalls wird der Wert aus Spalte V2 kopiert.
<code>TEMP = IF(ABS(V1-AVG(V1)) &lt; STDV(V1), V1)</code>	Erstellt eine neue Spalte TEMP, die jeden Wert in Spalte V1 enthält, der weniger als eine Standardabweichung vom Mittelwert entfernt ist.
<code>TEMP = IF(V3[20:30], V1[30:40], V2)</code>	Erstellt eine neue Spalte TEMP, die Werte für Zeile 10-20 enthält. Jeder Wert wird aus Spalte V1 (Zelle 10-20) kopiert, wenn der entsprechende Wert von Spalte V3 (Zelle 30-40) ungleich null ist; andernfalls wird der Wert aus Spalte V2 (Zelle 1-11) kopiert.

---

## IN-Makro

Das IN-Makro Makro ist in IBM Campaign und IBM Interact verfügbar.

### Syntax

Werte IN (Wert1 AND Wert2 . . . .) oder Werte IN Unterabfrage

### Parameter

Die erste Form ermöglicht die Verwendung einer Werteliste statt einer Unterabfrage.

Die zweite Form verwendet eine Unterabfrage, deren Auswertung ein Zwischenergebnis erzeugt, auf das weitere Verarbeitungsschritte angewendet werden können.

### Beschreibung

Das Prädikat IN ermöglicht die Verwendung einer Liste von Werten statt einer Unterabfrage oder es leitet eine Unterabfrage ein.

**Anmerkung:** Das Prädikat IN verfügt über eine negative Version, NOT IN. Deren Format ist mit dem von IN identisch. NOT IN ist nur wahr, wenn der angegebene Wert nicht in den von der Unterabfrage zurückgegebenen Werten gefunden wird.

**Wichtig:** Wenn Sie IN in IBM Interact einsetzen, können Sie nur die Syntax value IN (value1 AND value2 . . . .) verwenden.

### Beispiele

TEMP = IN(25, COLUMN(1...10))
Gibt die angegebene(n) Spalte(n) aus einem Datenbereich zurück.
TEMP = IN("cat", COLUMN("cat", "dog", "bird"))
Erstellt eine neue Spalte TEMP, die den Wert eins enthält.
TEMP = IN(V1, V1)
Erstellt eine neue Spalte TEMP, die nur Einsen enthält.
TEMP = IN(V1, V2)
Erstellt eine neue Spalte TEMP, in der jeder Wert eins ist, wenn die entsprechende Zeile von Spalte V1 einen Wert in Spalte V2 enthält; andernfalls ist der Wert null.

---

## INT-Makro

Das INT-Makro Makro ist nur in IBM Campaign verfügbar.

### Syntax

INT(data)

### Parameter

data

Die numerischen Werte, die auf einen Ganzzahlwert abgerundet werden sollen. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

## Beschreibung

INT berechnet die größte Ganzzahl, die kleiner als die Werte im angegebenen Datenbereich ist (gleichbedeutend mit der Untergrenze). INT gibt für jede Eingabespalte eine neue Spalte zurück, die die Untergrenze der Zahlen in der entsprechenden Eingabespalte enthält.

**Anmerkung:** INT ist mit der Makrofunktion FLOOR identisch.

## Beispiele

TEMP = INT(4.7)
Erstellt eine neue Spalte TEMP, die den Wert 4 enthält.
TEMP = INT(-1.5)
Erstellt eine neue Spalte TEMP, die den Wert -2 enthält.
TEMP = INT(V1)
Erstellt eine neue Spalte TEMP, in der jeder Wert die größte Ganzzahl kleiner-gleich dem Inhalt von Spalte V1 darstellt.
TEMP = V1 - INT(V1)
Erstellt eine neue Spalte TEMP, die den Dezimalanteil jedes Werts in Spalte V1 enthält.
TEMP = INT(V1:V3)
Erstellt drei neue Spalten TEMP, VX und VY. Die Werte in der Spalte TEMP sind jeweils die größten Ganzzahlen kleiner-gleich dem Inhalt von Spalte V1, die Werte von Spalte VX sind die größten Ganzzahlen kleiner-gleich dem Inhalt von Spalte V2 und die Werte von Spalte VY sind die größten Ganzzahlen kleiner-gleich dem Inhalt von Spalte V3.
TEMP = INT(V1[10:20])
Erstellt eine neue Spalte TEMP, in der die ersten 11 Zellen die größten Ganzzahlen kleiner-gleich den entsprechenden Werten in Zeile 10-20 von Spalte V1 darstellen. Die anderen Zellen in TEMP sind leer.
TEMP = INT(V1[1:5]:V2)
Erstellt zwei neue Spalten TEMP und VX, die jeweils die Werte in Zeile 1-5 enthalten (die anderen Zellen sind leer). Die Werte in der Spalte TEMP sind die größten Ganzzahlen kleiner-gleich den entsprechenden Zeilenwerten von Spalte V1, die Werte in der Spalte VX sind die größten Ganzzahlen kleiner-gleich den entsprechenden Zeilenwerten von Spalte V2.

## Zugehörige Funktionen

Funktion	Beschreibung
ROUND	Berechnet den gerundeten Wert des Inhalts des angegebenen Datenbereichs.
TRUNCATE	Berechnet den Ganzzahlanteil jedes Werts im angegebenen Datenbereich.

---

## INVERSE-Makro

Das INVERSE-Makro ist nur in IBM Campaign verfügbar.

### Syntax

INVERSE(data)

### Parameter

data

Die numerischen Werte, deren Umkehrfunktion berechnet werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

### Beschreibung

INVERSE berechnet den Negativwert der Werte im angegebenen Datenbereich. INVERSE gibt  $-x$  zurück (d. h., negative Werte werden als positive Werte zurückgegeben, positive Werte werden als negative Werte zurückgegeben). INVERSE gibt für jede Eingabespalte eine neue Spalte zurück, die die Umkehrfunktion der Werte in der entsprechenden Eingabespalte enthält.

**Anmerkung:** Stellen Sie zum Umkehren eines Werts oder einer Spalte ein Minuszeichen (-) voran. Beispielsweise ist  $V2 = -V1$  identisch mit  $V2 = \text{INVERSE}(V1)$ .

### Beispiele

TEMP = INVERSE(3.2)
Erstellt eine neue Spalte TEMP, die den Wert -3.2 enthält.
TEMP = INVERSE(V1)
Erstellt eine neue Spalte TEMP, in der jeder Wert den Negativwert der Werte in Spalte V1 darstellt.
TEMP = INVERSE(V1:V3)
Erstellt drei neue Spalten TEMP, VX und VY. Die Werte in der Spalte TEMP sind jeweils der Negativwert der Werte in Spalte V1, die Werte von Spalte VX sind der Negativwert der Werte in Spalte V2 und die Werte von Spalte VY sind der Negativwert der Werte in Spalte V3.
TEMP = INVERSE(V1[10:20])
Erstellt eine neue Spalte TEMP, in der die ersten 11 Zellen die negativen Werte der Werte in Zeile 10-20 von Spalte V1 darstellen. Die anderen Zellen in TEMP sind leer.
TEMP = INVERSE(V1[1:5]:V2)
Erstellt zwei neue Spalten TEMP und VX, die jeweils die Werte in Zeile 1-5 enthalten (die anderen Zellen sind leer). Die Werte in der Spalte TEMP sind die negativen Werte der entsprechenden Zeilen von Spalte V1, die Werte in der Spalte VX sind die negativen Werte der entsprechenden Zeilen von Spalte V2.

## Zugehörige Funktionen

Funktion	Beschreibung
ABS	Berechnet den absoluten Wert des Inhalts des angegebenen Datenbereichs.
NOT	Berechnet das logische Nicht des Inhalts des angegebenen Datenbereichs.
SIGN	Berechnet das Vorzeichen (positiv oder negativ) der Werte im angegebenen Datenbereich.

---

## IS-Makro

Das IS-Makro Makro ist in IBM Campaign und IBM Interact verfügbar.

### Syntax

IS <keyword>

### Parameter

keyword

Suchbedingung, üblicherweise "NULL," "TRUE," "UNKNOWN" oder "FALSE".

### Beschreibung

IS wird in komplexen Suchbedingungen eingesetzt. Je komplexer die Suche, desto nützlicher kann die IS-Bedingung sein. Diese booleschen Suchbedingungen stellen eine alternative Möglichkeit dar, einfache Suchbedingungen zu formulieren.

IS gibt in IBM Interact andere Ergebnisse zurück als in IBM Campaign. Bei NULL wird 1 zurückgegeben, wenn mindestens ein Nullwert für eine Zielgruppen-ID vorliegt. Bei UNKNOWN wird für eine Zielgruppen-ID 1 zurückgegeben, wenn sie keinen Wert aufweist.

---

## ISERROR-Makro

Das ISERROR-Makro Makro ist nur in IBM Campaign verfügbar.

### Syntax

ISERROR(data)

### Parameter

data

Die zu prüfenden Werte, wenn eine der Zeilen einen Fehler (d. h. eine Zelle mit ??? enthält. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makro funktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

## Beschreibung

ISERROR prüft, ob eine der Zellen in jeder Zeile des angegebenen Datenbereichs einen Fehler (d. h. eine Zelle mit ??? enthält. ISERROR gibt eine neue Spalte zurück, in der jede Zeile eine Eins enthält, wenn die entsprechende Zeile von data einen Fehler enthält. Andernfalls enthält die Zeile eine Null. Diese zeilenweise ausgeführte Berechnung erzeugt für jede Zeile ein Ergebnis, bis zum letzten Wert der längsten Spalte.

## Beispiele

TEMP = ISERROR(-3)
Erstellt eine neue Spalte TEMP, die den Wert null enthält.
TEMP = ISERROR(V1)
Erstellt eine neue Spalte TEMP, in der jeder Wert eins ist, wenn die entsprechende Zeile von Spalte V1 die Angabe ??? enthält; andernfalls ist der Wert null.
TEMP = ISERROR(V1:V3)
Erstellt eine neue Spalte TEMP, in der jeder Wert eins ist, wenn eine der Zellen in den entsprechenden Zeilen von Spalte V1 bis V3 die Angabe ??? enthält; andernfalls ist der Wert null.
TEMP = ISERROR(V1[50:100]:V10)
Erstellt eine neue Spalte TEMP mit Werten in Zeile 1-50. Jeder Wert ist eins, wenn eine der Zellen in Zeile 50-100 von Spalte V1 bis V10 die Angabe ??? enthält; andernfalls ist der Wert null.

---

## ISODD-Makro

Das ISODD-Makro Makro ist nur in IBM Campaign verfügbar.

### Syntax

ISODD(data)

### Parameter

data

Die numerischen Werte, bei denen geprüft werden soll, ob sie ungerade sind. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

### Beschreibung

ISODD testet jeden Wert im angegebenen Dataset auf Ungeradheit. ISODD gibt für jede Eingabespalte eine neue Spalte zurück, die bei allen ungeraden Werten (d. h., der Wert modulo zwei ist eins) eine Eins und bei allen geraden Werten eine Null enthält.

**Anmerkung:** Bei Werten, die keine Ganzzahlen sind, wird zunächst die Makrofunktion INT angewendet. Beispiel: ISODD(2.5) = 0, da 2 nicht ungerade ist.

## Beispiele

TEMP = ISODD(-3)
Erstellt eine neue Spalte TEMP, die den Wert eins enthält.
TEMP = ISODD(V1)
Erstellt eine neue Spalte TEMP, in der jeder Wert das Ergebnis der Prüfung des Inhalts von Spalte V1 auf Ungeradheit darstellt.
TEMP = ISODD(V1:V3)
Erstellt drei neue Spalten TEMP, VX und VY. Die Werte in der Spalte TEMP sind das Ergebnis der Prüfung des Inhalts von Spalte V1 auf Ungeradheit, die Werte von Spalte VX sind das Ergebnis der Prüfung des Inhalts von Spalte V2 auf Ungeradheit und die Werte von Spalte VY sind das Ergebnis der Prüfung des Inhalts von Spalte V3 auf Ungeradheit.
TEMP = ISODD(V1[10:20])
Erstellt eine neue Spalte TEMP, in der die ersten 11 Zellen die Ergebnisse der Prüfung der Werte in Zeile 10-20 von Spalte V1 auf Ungeradheit enthalten. Die anderen Zellen in TEMP sind leer.
TEMP = ISODD(V1[1:5]:V2)
Erstellt zwei neue Spalten TEMP und VX, die jeweils die Werte in Zeile 1-5 enthalten (die anderen Zellen sind leer). Die Werte in der Spalte TEMP sind die Ergebnisse der Prüfung der entsprechenden Zeilen von Spalte V1 auf Ungeradheit, die Werte in der Spalte VX sind die Ergebnisse der Prüfung der entsprechenden Zeilen von Spalte V2 auf Ungeradheit.

## Zugehörige Funktionen

Funktion	Beschreibung
ISEVEN	Prüft, ob die Eingabewerte gerade (d. h. durch zwei teilbar) sind

---

## ISEVEN-Makro

Das ISEVEN-Makro Makro ist nur in IBM Campaign verfügbar.

### Syntax

ISEVEN(data)

### Parameter

data

Die numerischen Werte, bei denen geprüft werden soll, ob sie gerade sind. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

## Beschreibung

ISEVEN testet jeden Wert im angegebenen Dataset auf Geradheit. ISEVEN gibt für jede Eingabespalte eine neue Spalte zurück, die bei allen geraden Werten (d. h., der Wert modulo zwei ist null) eine Eins und bei allen ungeraden Werten eine Null enthält.

**Anmerkung:** Bei Werten, die keine Ganzzahlen sind, wird zunächst die Makrofunktion INT angewendet. Beispiel: ISEVEN(2.5) = 1, da 2 gerade ist.

## Beispiele

TEMP = ISEVEN(-3)
Erstellt eine neue Spalte TEMP, die den Wert null enthält.
TEMP = ISEVEN(V1)
Erstellt eine neue Spalte TEMP, in der jeder Wert das Ergebnis der Prüfung des Inhalts von Spalte V1 auf Geradheit darstellt.
TEMP = ISEVEN(V1:V3)
Erstellt drei neue Spalten TEMP, VX und VY. Die Werte in der Spalte TEMP sind das Ergebnis der Prüfung des Inhalts von Spalte V1 auf Geradheit, die Werte von Spalte VX sind das Ergebnis der Prüfung des Inhalts von Spalte V2 auf Geradheit und die Werte von Spalte VY sind das Ergebnis der Prüfung des Inhalts von Spalte V3 auf Geradheit.
TEMP = ISEVEN(V1[10:20])
Erstellt eine neue Spalte TEMP, in der die ersten 11 Zellen die Ergebnisse der Prüfung der Werte in Zeile 10-20 von Spalte V1 auf Geradheit enthalten. Die anderen Zellen in TEMP sind leer.
TEMP = ISEVEN(V1[1:5]:V2)
Erstellt zwei neue Spalten TEMP und VX, die jeweils die Werte in Zeile 1-5 enthalten (die anderen Zellen sind leer). Die Werte in der Spalte TEMP sind die Ergebnisse der Prüfung der entsprechenden Zeilen von Spalte V1 auf Geradheit, die Werte in der Spalte VX sind die Ergebnisse der Prüfung der entsprechenden Zeilen von Spalte V2 auf Geradheit.

## Zugehörige Funktionen

Funktion	Beschreibung
ISODD	Prüft, ob die Eingabewerte ungerade (d. h. nicht durch zwei teilbar) sind

---

## ISODD-Makro

Das ISODD-Makro Makro ist nur in IBM Campaign verfügbar.

### Syntax

ISODD(data)

### Parameter

data

Die numerischen Werte, bei denen geprüft werden soll, ob sie ungerade sind. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

## Beschreibung

ISODD testet jeden Wert im angegebenen Dataset auf Ungeradheit. ISODD gibt für jede Eingabespalte eine neue Spalte zurück, die bei allen ungeraden Werten (d. h., der Wert modulo zwei ist eins) eine Eins und bei allen geraden Werten eine Null enthält.

**Anmerkung:** Bei Werten, die keine Ganzzahlen sind, wird zunächst die Makrofunktion INT angewendet. Beispiel: ISODD(2.5) = 0, da 2 nicht ungerade ist.

## Beispiele

TEMP = ISODD(-3)
Erstellt eine neue Spalte TEMP, die den Wert eins enthält.
TEMP = ISODD(V1)
Erstellt eine neue Spalte TEMP, in der jeder Wert das Ergebnis der Prüfung des Inhalts von Spalte V1 auf Ungeradheit darstellt.
TEMP = ISODD(V1:V3)
Erstellt drei neue Spalten TEMP, VX und VY. Die Werte in der Spalte TEMP sind das Ergebnis der Prüfung des Inhalts von Spalte V1 auf Ungeradheit, die Werte von Spalte VX sind das Ergebnis der Prüfung des Inhalts von Spalte V2 auf Ungeradheit und die Werte von Spalte VY sind das Ergebnis der Prüfung des Inhalts von Spalte V3 auf Ungeradheit.
TEMP = ISODD(V1[10:20])
Erstellt eine neue Spalte TEMP, in der die ersten 11 Zellen die Ergebnisse der Prüfung der Werte in Zeile 10-20 von Spalte V1 auf Ungeradheit enthalten. Die anderen Zellen in TEMP sind leer.
TEMP = ISODD(V1[1:5]:V2)
Erstellt zwei neue Spalten TEMP und VX, die jeweils die Werte in Zeile 1-5 enthalten (die anderen Zellen sind leer). Die Werte in der Spalte TEMP sind die Ergebnisse der Prüfung der entsprechenden Zeilen von Spalte V1 auf Ungeradheit, die Werte in der Spalte VX sind die Ergebnisse der Prüfung der entsprechenden Zeilen von Spalte V2 auf Ungeradheit.

## Zugehörige Funktionen

Funktion	Beschreibung
ISEVEN	Prüft, ob die Eingabewerte gerade (d. h. durch zwei teilbar) sind

---

## LE-Makro

Das LE-Makro Makro ist in IBM Campaign und IBM Interact verfügbar.

## Syntax

```
data1 LE data2 data1 <= data2
```

## Parameter

data1

Der numerische Zellenbereich, der verglichen werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

data2

Die Zahlen, mit denen alle Werte in der angegebenen Spalte verglichen werden sollen. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

## Beschreibung

LE vergleicht die beiden angegebenen Datenbereiche und gibt eine Eins zurück, wenn die Werte im ersten Dataset kleiner-gleich den Werten im zweiten Dataset sind, andernfalls eine Null. LE gibt für jede Eingabespalte eine neue Spalte zurück, wobei jeweils die entsprechende Spalte von data1 mit der entsprechenden Spalte von data2 verglichen wird (d. h., die erste Spalte von data1 wird mit der ersten Spalte von data2 verglichen, die zweite Spalte mit der zweiten Spalte usw.).

Wenn es sich bei data2 um eine Konstante handelt, wird jeder Wert in data1 mit dem betreffenden Wert verglichen. Wenn es sich bei data2 um eine Spalte handelt, werden die Berechnungen zeilenweise ausgeführt. Die erste Zeile von data1 wird mit der ersten Zeile von data2 verglichen, die zweite Zeile mit der zweiten Zeile usw. Diese zeilenweise ausgeführte Berechnung erzeugt für jede Zeile ein Ergebnis, bis zum letzten Wert der kürzesten Spalte.

**Anmerkung:** Der Operator LE kann durch ein Kleiner-als-Zeichen, auf das ein Gleichheitszeichen folgt, abgekürzt werden (<=).

## Beispiele

```
TEMP = 4 LE 4 oder TEMP = 4 <= 4
```

Erstellt eine neue Spalte TEMP, die den Wert eins enthält (vier ist gleich sich selbst).

```
TEMP = V1 <= 8
```

Erstellt eine neue Spalte TEMP, in der jeder Wert eins ist, wenn der entsprechende Zeilenwert von Spalte V1 kleiner-gleich der Zahl acht ist; andernfalls ist der Wert null.

```
TEMP = V1:V3 <= 2
```

Erstellt drei neue Spalten TEMP, VX und VY. Die Werte in der Spalte TEMP sind der mit zwei verglichene Inhalt von Spalte V1, die Werte von Spalte VX sind der mit zwei verglichene Inhalt von Spalte V2 und die Werte von Spalte VY sind der mit zwei verglichene Inhalt von Spalte V3.

TEMP = V1 <= V1
Erstellt eine neue Spalte TEMP, die nur Einsen enthält (jede Zahl ist gleich sich selbst).
TEMP = V1 <= V2
Erstellt eine neue Spalte TEMP, wobei jeder Zeilenwert von Spalte V1 mit dem entsprechenden Zeilenwert von Spalte V2 verglichen wird.
TEMP = V1[10:20] <= V2 oder TEMP = V1[10:20] <= V2[1:11]
Erstellt eine neue Spalte TEMP, deren erste 11 Zellen das Ergebnis des Vergleichs der Werte in Zeile 10-20 von Spalte V1 mit den Werten in Zeile 1-11 von Spalte V2 enthalten. Die anderen Zellen in TEMP sind leer.

## Zugehörige Funktionen

Funktion	Beschreibung
EQ	Gibt TRUE zurück, wenn der eine Datenbereich dem anderen gleich ist.
GE	Gibt TRUE zurück, wenn der eine Datenbereich größer-gleich dem anderen ist.
GT	Gibt TRUE zurück, wenn der eine Datenbereich größer als der andere ist.
LT	Gibt TRUE zurück, wenn der eine Datenbereich kleiner als der andere ist.
NE	Gibt TRUE zurück, wenn der eine Datenbereich dem anderen ungleich ist.

---

## LIKE-Makro

Das LIKE-Makro Makro ist in IBM Campaign und IBM Interact verfügbar.

### Syntax

data1 [NOT] LIKE data2

### Parameter

data1

Der Zellenbereich, der verglichen werden soll. Dabei kann es sich um eine Zeichenfolge oder um einen Ausdruck handeln, dessen Auswertung eine Zeichenfolge ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

data2

Das Textmuster, mit dem alle Werte in der angegebenen Spalte verglichen werden sollen. Dabei kann es sich um eine Zeichenfolge oder um einen Ausdruck handeln, dessen Auswertung eine Zeichenfolge ergibt. Die Anzahl der Spalten in data2 muss mit der Anzahl der Spalten in data1 übereinstimmen, es sei denn, bei data2

handelt es sich um eine Konstante. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

Ein Unterstrich (\_) in data2 stellt ein Platzhalterzeichen dar, das mit einem beliebigen einzelnen Zeichen in data1 übereinstimmt. Ein Prozentzeichen (%) stimmt mit null oder mehr Zeichen in data1 überein.

## Beschreibung

LIKE vergleicht die beiden angegebenen Datenbereiche und gibt eine Eins zurück, wenn die Zeichenfolgen übereinstimmen, bzw. eine Null, wenn die Zeichenfolgen nicht übereinstimmen. LIKE gibt für jede Eingabespalte eine neue Spalte zurück, wobei jeweils die entsprechende Spalte von data1 mit der entsprechenden Spalte von data2 verglichen wird (d. h., die erste Spalte von data1 wird mit der ersten Spalte von data2 verglichen, die zweite Spalte mit der zweiten Spalte usw.).

Wenn es sich bei data2 um eine Zeichenfolgekonstante handelt, wird jede Zeichenfolge in data1 mit der betreffenden Zeichenfolge verglichen. Wenn es sich bei data2 um eine Spalte handelt, werden die Berechnungen zeilenweise ausgeführt. Die Zeichenfolge der ersten Zeile von data1 wird mit der Zeichenfolge der ersten Zeile von data2 verglichen, die Zeichenfolge der zweiten Zeile mit der Zeichenfolge der zweiten Zeile usw. Diese zeilenweise ausgeführte Berechnung erzeugt für jede Zeile ein Ergebnis, bis zur letzten Zeichenfolge der kürzesten Spalte.

Beim Vergleich von Zeichenfolgen spielt die Groß-/Kleinschreibung keine Rolle (d. h., "Ja", "ja", "JA" und "jA" werden gleich behandelt).

**Anmerkung:** Das Makro LIKE verfügt über eine negative Version, NOT LIKE. Deren Format ist mit dem von LIKE identisch. NOT LIKE gibt eine Eins zurück, wenn die Zeichenfolge in data1 nicht mit der durch data2 definierten Schablone übereinstimmt.

## Beispiele

TEMP = "gold" LIKE "gold"
Erstellt eine neue Spalte TEMP, die den Wert eins enthält (die beiden Zeichenfolgen stimmen überein).
TEMP = "No" LIKE "NO"
Erstellt eine neue Spalte TEMP, die den Wert eins enthält (beim Vergleich von Zeichenfolgen wird die Groß-/Kleinschreibung nicht beachtet).
TEMP = V1 LIKE "gold%"
Erstellt eine neue Spalte TEMP, in der jeder Wert eins ist, wenn der entsprechende Zeilenwert von Spalte V1 mit der Zeichenfolge "Gold", auf die eine beliebige Anzahl von Zeichen folgt, übereinstimmt. Andernfalls ist jeder Wert null.
TEMP = V1 LIKE "g_ld"
Erstellt eine neue Spalte TEMP, in der jeder Wert eins ist, wenn der entsprechende Zeilenwert von Spalte V1 mit der Zeichenfolge aus "G", einem beliebigen Zeichen und den Zeichen "ld" übereinstimmt. Andernfalls ist jeder Wert null.
TEMP = V1 LIKE V1
Erstellt eine neue Spalte TEMP, die nur Einsen enthält (jede Zahl ist gleich sich selbst).

TEMP = V1 LIKE V2
Erstellt eine neue Spalte TEMP, wobei jeder Zeilenwert von Spalte V1 mit dem entsprechenden Zeilenwert von Spalte V2 verglichen wird.
TEMP = V1:V3 LIKE V4:V6
Erstellt drei neue Spalten TEMP, VX und VY. In der Spalte TEMP sind die Zeichenfolgen von Spalte V1 mit den entsprechenden Zeichenfolgen von Spalte V4 verglichen. Die Spalte VX enthält den Vergleich von Spalte V2 mit Spalte V5. Die Spalte VY enthält den Vergleich von Spalte V3 mit Spalte V6.
TEMP = V1[10:20] LIKE V2 oder TEMP = V1[10:20] LIKE V2[1:11]
Erstellt eine neue Spalte TEMP, deren erste 11 Zellen das Ergebnis des Vergleichs der Zeichenfolgen in Zeile 10-20 von Spalte V1 mit den Zeichenfolgen in Zeile 1-11 von Spalte V2 enthalten. Die anderen Zellen in TEMP sind leer.

## Zugehörige Funktionen

Funktion	Beschreibung
EQ	Gibt TRUE zurück, wenn der eine Datenbereich dem anderen gleich ist.
GE	Gibt TRUE zurück, wenn der eine Datenbereich größer-gleich dem anderen ist.
GT	Gibt TRUE zurück, wenn der eine Datenbereich größer als der andere ist.
LE	Gibt TRUE zurück, wenn der eine Datenbereich kleiner-gleich dem anderen ist.
LT	Gibt TRUE zurück, wenn der eine Datenbereich kleiner als der andere ist.
NE	Gibt TRUE zurück, wenn der eine Datenbereich dem anderen ungleich ist.

---

## LN- oder LOG-Makro

Das LN- oder LOG-Makro Makro ist nur in IBM Campaign verfügbar.

### Syntax

LN(data) oder LOG(data)

### Parameter

data

Die numerischen Werte, deren natürlicher Logarithmus berechnet werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

## Beschreibung

LN oder LOG berechnet den natürlichen Logarithmus jedes Wertes im angegebenen Datenbereich. Dabei wird für jede Eingabespalte eine neue Spalte zurückgegeben, die den natürlichen Logarithmus der Zahlen in der entsprechenden Eingabespalte enthält. Natürliche Logarithmen basieren auf der Konstante  $e = 2,7182818$ . LN ist die Umkehrfunktion der Makrofunktion EXP.

**Anmerkung:** Alle Werte im angegebenen Datenbereich müssen größer als null sein. Andernfalls wird für jede ungültige Eingabe eine leere Zelle zurückgegeben.

## Beispiele

TEMP = LN(3) oder TEMP = LOG(3)
Erstellt eine neue Spalte TEMP, die den Wert 1.099 enthält.
TEMP = LN(V1)
Erstellt eine neue Spalte TEMP, in der jeder Wert den natürlichen Logarithmus des Inhalts von Spalte V1 darstellt.
TEMP = LN(V1:V3)
Erstellt drei neue Spalten TEMP, VX und VY. Die Werte in der Spalte TEMP sind jeweils der natürliche Logarithmus des Inhalts von Spalte V1, die Werte von Spalte VX sind der natürliche Logarithmus des Inhalts von Spalte V2 und die Werte von Spalte VY sind der natürliche Logarithmus des Inhalts von Spalte V3.
TEMP = LN(V1[10:20])
Erstellt eine neue Spalte TEMP, in der die ersten 11 Zellen jeweils den natürlichen Logarithmus der Werte in Zeile 10-20 von Spalte V1 darstellen. Die anderen Zellen in TEMP sind leer.
TEMP = LN(V1[1:5]:V2)
Erstellt zwei neue Spalten TEMP und VX, die jeweils die Werte in Zeile 1-5 enthalten (die anderen Zellen sind leer). Die Werte in der Spalte TEMP sind der natürliche Logarithmus der entsprechenden Zeilen von Spalte V1, die Werte in der Spalte VX sind der natürliche Logarithmus der entsprechenden Zeilen von Spalte V2.

## Zugehörige Funktionen

Funktion	Beschreibung
EXP	Erhebt die natürliche Zahl (e) in die Potenz, die durch den Inhalt jeder Zelle im angegebenen Datenbereich angegeben ist.
LOG2	Berechnet den binären Logarithmus des Inhalts des angegebenen Datenbereichs.
LOG10	Berechnet den dekadischen Logarithmus des Inhalts des angegebenen Datenbereichs.
POW	Erhebt einen Basiswert in die Potenz des oder der angegebenen Exponenten.

---

## LOG2-Makro

Das LOG2-Makro Makro ist nur in IBM Campaign verfügbar.

## Syntax

LOG2(data)

## Parameter

data

Die numerischen Werte, deren binärer Logarithmus berechnet werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

## Beschreibung

LOG2 berechnet den binären Logarithmus der Werte im angegebenen Datenbereich. Dabei wird für jede Eingabespalte eine neue Spalte zurückgegeben, die den binären Logarithmus der Zahlen in der entsprechenden Eingabespalte enthält.

**Anmerkung:** Alle Werte im angegebenen Datenbereich müssen größer als null sein. Andernfalls wird für jede ungültige Eingabe eine leere Zelle zurückgegeben.

## Beispiele

TEMP = LOG2(8)
Erstellt eine neue Spalte TEMP, die den Wert drei enthält.
TEMP = LOG2(V1)
Erstellt eine neue Spalte TEMP, in der jeder Wert den binären Logarithmus des Inhalts von Spalte V1 darstellt.
TEMP = LOG2(V1:V3)
Erstellt drei neue Spalten TEMP, VX und VY. Die Werte in der Spalte TEMP sind jeweils der binäre Logarithmus des Inhalts von Spalte V1, die Werte von Spalte VX sind der binäre Logarithmus des Inhalts von Spalte V2 und die Werte von Spalte VY sind der binäre Logarithmus des Inhalts von Spalte V3.
TEMP = LOG2(V1[10:20])
Erstellt eine neue Spalte TEMP, in der die ersten 11 Zellen jeweils den binären Logarithmus der Werte in Zeile 10-20 von Spalte V1 darstellen. Die anderen Zellen in TEMP sind leer.
TEMP = LOG2(V1[1:5]:V2)
Erstellt zwei neue Spalten TEMP und VX, die jeweils die Werte in Zeile 1-5 enthalten (die anderen Zellen sind leer). Die Werte in der Spalte TEMP sind der binäre Logarithmus der entsprechenden Zeilen von Spalte V1, die Werte in der Spalte VX sind der binäre Logarithmus der entsprechenden Zeilen von Spalte V2.

## Zugehörige Funktionen

Funktion	Beschreibung
LN oder LOG	Berechnet den natürlichen Logarithmus des Inhalts des angegebenen Datenbereichs.
LOG10	Berechnet den dekadischen Logarithmus des Inhalts des angegebenen Datenbereichs.

Funktion	Beschreibung
POW	Erhebt einen Basiswert in die Potenz des angegebenen Exponenten.

## LOG10-Makro

Das LOG10-Makro Makro ist nur in IBM Campaign verfügbar.

### Syntax

LOG10(data)

### Parameter

data

Die numerischen Werte, deren dekadischer Logarithmus berechnet werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

### Beschreibung

LOG10 berechnet den dekadischen Logarithmus der Werte im angegebenen Datenbereich. Dabei wird für jede Eingabespalte eine neue Spalte zurückgegeben, die den dekadischen Logarithmus der Zahlen in der entsprechenden Eingabespalte enthält.

**Anmerkung:** Alle Werte im angegebenen Datenbereich müssen größer als null sein. Andernfalls wird für jede ungültige Eingabe eine leere Zelle zurückgegeben.

### Beispiele

TEMP = LOG10(100)
Erstellt eine neue Spalte TEMP, die den Wert zwei enthält.
TEMP = LOG10(V1)
Erstellt eine neue Spalte TEMP, in der jeder Wert den dekadischen Logarithmus des Inhalts von Spalte V1 darstellt.
TEMP = LOG10(V1:V3)
Erstellt drei neue Spalten TEMP, VX und VY. Die Werte in der Spalte TEMP sind jeweils der dekadische Logarithmus des Inhalts von Spalte V1, die Werte von Spalte VX sind der dekadische Logarithmus des Inhalts von Spalte V2 und die Werte von Spalte VY sind der dekadische Logarithmus des Inhalts von Spalte V3.
TEMP = LOG10(V1[10:20])
Erstellt eine neue Spalte TEMP, in der die ersten 11 Zellen jeweils den dekadischen Logarithmus der Werte in Zeile 10-20 von Spalte V1 darstellen. Die anderen Zellen in TEMP sind leer.

TEMP = LOG10(V1[1:5]:V2)

Erstellt zwei neue Spalten TEMP und VX, die jeweils die Werte in Zeile 1-5 enthalten (die anderen Zellen sind leer). Die Werte in der Spalte TEMP sind der dekadische Logarithmus der entsprechenden Zeilen von Spalte V1, die Werte in der Spalte VX sind der dekadische Logarithmus der entsprechenden Zeilen von Spalte V2.

## Zugehörige Funktionen

Funktion	Beschreibung
LN oder LOG	Berechnet den natürlichen Logarithmus des Inhalts des angegebenen Datenbereichs.
LOG2	Berechnet den binären Logarithmus des Inhalts des angegebenen Datenbereichs.
POW	Erhebt einen Basiswert in die Potenz des angegebenen Exponenten.

---

## LOWER-Makro

Das LOWER-Makro Makro ist in IBM Campaign und IBM Interact verfügbar.

### Syntax

LOWER(data)

### Parameter

data

Der Zeichenfolgewart, der in Kleinbuchstaben konvertiert werden soll.

### Beschreibung

LOWER konvertiert jeden Zeichenfolgewart im angegebenen Datenbereich in Kleinbuchstaben. Dabei wird eine neue Spalte zurückgegeben, in der jede Zelle die in Kleinbuchstaben umgewandelte Zeichenfolge der entsprechenden Eingangszelle enthält.

### Beispiele

Temp = LOWER "GOLD"

Erstellt eine neue Spalte TEMP, die den Wert "gold" enthält.

TEMP = LOWER( "JAN 15, 1997")

Erstellt eine neue Spalte TEMP, die die ASCII-Zeichenfolge "jan 15, 1997" enthält.

TEMP = LOWER( "Pressure")

Erstellt eine neue Spalte TEMP, die die ASCII-Zeichenfolge "pressure" enthält.

TEMP = LOWER(V1)

Erstellt eine neue Spalte TEMP, die die Kleinbuchstaben jeder Zeichenfolge in der Spalte V1 enthält.

---

## LT-Makro

Das LT-Makro Makro ist in IBM Campaign und IBM Interact verfügbar.

### Syntax

```
data1 LT data2 data1 < data2
```

### Parameter

data1

Der numerische Zellenbereich, der verglichen werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

data2

Die Zahlen, mit denen alle Werte in der angegebenen Spalte verglichen werden sollen. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

### Beschreibung

LT vergleicht die beiden angegebenen Datenbereiche und gibt eine Eins zurück, wenn die Werte im ersten Dataset kleiner als die Werte im zweiten Dataset sind, andernfalls eine Null. LT gibt für jede Eingabespalte eine neue Spalte zurück, wobei jeweils die entsprechende Spalte von data1 mit der entsprechenden Spalte von data2 verglichen wird (d. h., die erste Spalte von data1 wird mit der ersten Spalte von data2 verglichen, die zweite Spalte mit der zweiten Spalte usw.).

Wenn es sich bei data2 um eine Konstante handelt, wird jeder Wert in data1 mit dem betreffenden Wert verglichen. Wenn es sich bei data2 um eine Spalte handelt, werden die Berechnungen zeilenweise ausgeführt. Die erste Zeile von data1 wird mit der ersten Zeile von data2 verglichen, die zweite Zeile mit der zweiten Zeile usw. Diese zeilenweise ausgeführte Berechnung erzeugt für jede Zeile ein Ergebnis, bis zum letzten Wert der kürzesten Spalte.

**Anmerkung:** Der Operator LT kann durch ein Kleiner-als-Zeichen abgekürzt werden (<).

### Beispiele

```
TEMP = 3 LT 4 oder TEMP = 3 < 4
```

Erstellt eine neue Spalte TEMP, die den Wert eins enthält (drei ist kleiner als vier).

```
TEMP = V1 < 8
```

Erstellt eine neue Spalte TEMP, in der jeder Wert eins ist, wenn der entsprechende Zeilenwert von Spalte V1 kleiner als die Zahl acht ist; andernfalls ist der Wert null.

TEMP = V1:V3 < 2
Erstellt drei neue Spalten TEMP, VX und VY. Die Werte in der Spalte TEMP sind der mit zwei verglichene Inhalt von Spalte V1, die Werte von Spalte VX sind der mit zwei verglichene Inhalt von Spalte V2 und die Werte von Spalte VY sind der mit zwei verglichene Inhalt von Spalte V3.
TEMP = V1 < V1
Erstellt eine neue Spalte TEMP, die nur Nullen enthält (keine Zahl ist kleiner als sie selbst).
TEMP = V1 < V2
Erstellt eine neue Spalte TEMP, wobei jeder Zeilenwert von Spalte V1 mit dem entsprechenden Zeilenwert von Spalte V2 verglichen wird.
TEMP = V1[10:20] < V2 oder TEMP = V1[10:20] < V2[1:11]
Erstellt eine neue Spalte TEMP, deren erste 11 Zellen das Ergebnis des Vergleichs der Werte in Zeile 10-20 von Spalte V1 mit den Werten in Zeile 1-11 von Spalte V2 enthalten. Die anderen Zellen in TEMP sind leer.

## Zugehörige Funktionen

Funktion	Beschreibung
EQ	Gibt TRUE zurück, wenn der eine Datenbereich dem anderen gleich ist.
GE	Gibt TRUE zurück, wenn der eine Datenbereich größer-gleich dem anderen ist.
GT	Gibt TRUE zurück, wenn der eine Datenbereich größer als der andere ist.
LE	Gibt TRUE zurück, wenn der eine Datenbereich kleiner-gleich dem anderen ist.

---

## LTRIM-Makro

Das LTRIM-Makro Makro ist in IBM Campaign und IBM Interact verfügbar.

### Syntax

LTRIM(data)

### Parameter

data

Die Zeichenfolge, aus der führende Leerzeichen entfernt werden sollen.

### Beschreibung

LTRIM entfernt führende Leerzeichen aus jedem Zeichenfolgewert im angegebenen Datenbereich und gibt die konvertierte Zeichenfolge zurück. LTRIM gibt für jede Eingabespalte eine neue Spalte zurück.

## Beispiele

```
Temp = LTRIM " gold"
```

Erstellt eine neue Zeichenfolge Temp, die "gold" enthält.

---

## MAX-Makro

Das MAX-Makro Makro ist in IBM Campaign und IBM Interact verfügbar.

### Syntax

```
MAX(data [, keyword])
```

### Parameter

data

Die numerischen Werte, deren Maximum berechnet werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

keyword

Dieses optionale Schlüsselwort legt fest, wie die Berechnung auf den Eingabedatenbereich angewendet wird. Wählen Sie eine der folgenden Angaben aus:

ALL - Wendet die Berechnung auf alle Zellen in data an (Standard)

COL - Führt die Berechnung für jede Spalte von data gesondert aus

ROW - Führt die Berechnung für jede Zeile von data gesondert aus

Weitere Informationen zur Verwendung von Schlüsselwörtern in IBM Campaign finden Sie in „Formatspezifikationen“ auf Seite 9.

**Anmerkung:** Viele Makrofunktionen setzen die Schlüsselwortparameter {ALL | COL | ROW} ein. Diese Schlüsselwörter gelten nicht für IBM Campaign, da es sich bei den Eingabedaten immer um eine Einzelspalte oder ein Einzelfeld handelt. Das Makro verhält sich immer so, als ob das Schlüsselwort COL angegeben würde. Deshalb brauchen Sie diese Schlüsselwörter nicht anzugeben, wenn Sie IBM Campaign verwenden.

### Beschreibung

MAX berechnet das Maximum der Werte im angegebenen Datenbereich. Es wird eine einzelne neue Spalte zurückgegeben, die den Maximalwert enthält.

### Beispiele

```
TEMP = MAX(3) oder TEMP = MAX(3, ALL)
```

Erstellt eine neue Spalte TEMP, die den Wert drei enthält.

TEMP = MAX(V1)
Erstellt eine neue Spalte TEMP, die einen Einzelwert enthält, der den Maximalwert des Inhalts von Spalte V1 darstellt.
TEMP = MAX(V1:V3)
Erstellt eine neue Spalte TEMP, die einen Einzelwert enthält, der das Maximum des Inhalts der Spalten V1, V2 und V3 darstellt.
TEMP = MAX(V1[10:20])
Erstellt eine neue Spalte TEMP, die einen Einzelwert enthält, der das Maximum der Zellen in Zeile 10-20 von Spalte V1 darstellt.
TEMP = MAX(V1[1:5]:V4)
Erstellt eine neue Spalte TEMP, die einen Einzelwert enthält, der das Maximum der Zellen in Zeile 1-5 von Spalte V1 bis V4 darstellt.
TEMP = MAX(V1:V3, COL)
Erstellt drei neue Spalten TEMP, VX und VY. Der Einzelwert in der Spalte TEMP ist das Maximum des Inhalts von Spalte V1, der Einzelwert in der Spalte VX ist das Maximum des Inhalts von Spalte V2 und der Einzelwert in der Spalte VY ist das Maximum des Inhalts von Spalte V3.
TEMP = MAX(V1[1:5]:V3, COL)
Erstellt drei neue Spalten TEMP, VX und VY, die jeweils einen Einzelwert enthalten. Der Wert in der Spalte TEMP ist das Maximum der Zellen in Zeile 1-5 von Spalte V1, der Wert in der Spalte VX ist das Maximum der Zellen in Zeile 1-5 von Spalte V2 und der Wert in der Spalte VY ist das Maximum der Zellen in Zeile 1-5 von Spalte V3.
TEMP = MAX(V1:V3, ROW)
Erstellt eine neue Spalte TEMP, in der jeder Zelleneintrag das Maximum der entsprechenden Zeile in Spalte V1, V2 und V3 darstellt.
TEMP = MAX(V1[10:20]:V3, ROW)
Erstellt eine neue Spalte TEMP, in der die ersten 11 Zellen das Maximum der Werte in Zeile 10-20 von Spalte V1 bis V3 enthalten. Die anderen Zellen in TEMP sind leer.

## Zugehörige Funktionen

Funktion	Beschreibung
MIN	Berechnet das Minimum eines Zellenbereichs

---

## MEAN-Makro

Das MEAN-Makro Makro ist in IBM Campaign und IBM Interact verfügbar.

### Syntax

MEAN(data [, keyword])

### Parameter

data

Die numerischen Werte, deren arithmetisches Mittel berechnet werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

keyword

Dieses optionale Schlüsselwort legt fest, wie die Berechnung auf den Eingabedatenbereich angewendet wird. Wählen Sie eine der folgenden Angaben aus:

ALL - Wendet die Berechnung auf alle Zellen in data an (Standard)

COL - Führt die Berechnung für jede Spalte von data gesondert aus

ROW - Führt die Berechnung für jede Zeile von data gesondert aus

Weitere Informationen zur Verwendung von Schlüsselwörtern finden Sie unter „DATE“ auf Seite 52.

**Anmerkung:** Viele Makrofunktionen setzen die Schlüsselwortparameter {ALL | COL | ROW} ein. Diese Schlüsselwörter gelten nicht für IBM Campaign, da es sich bei den Eingabedaten immer um eine Einzelspalte oder ein Einzelfeld handelt. Das Makro verhält sich immer so, als ob das Schlüsselwort COL angegeben würde. Deshalb brauchen Sie diese Schlüsselwörter nicht anzugeben, wenn Sie IBM Campaign verwenden.

## Beschreibung

MEAN berechnet das arithmetische Mittel oder den Durchschnitt der Zellen im angegebenen Datenbereich. Das arithmetische Mittel wird berechnet, indem der Inhalt aller Zellen addiert und durch die Anzahl der Zellen geteilt wird. Die Anzahl der von MEAN zurückgegebenen Spalten hängt von keyword ab.

- Wenn keyword den Wert ALL hat, gibt MEAN eine neue Spalte zurück, die einen Einzelwert (den Durchschnitt aller Zellen in data) enthält.
- Wenn keyword den Wert COL hat, gibt MEAN für jede Eingabespalte eine neue Spalte zurück. Jede neue Spalte enthält einen Wert (den Durchschnitt aller Zellen in der entsprechenden Eingabespalte).
- Wenn keyword den Wert ROW hat, gibt MEAN eine neue Spalte zurück, die den Durchschnitt für jede Zeile von data enthält.

**Anmerkung:** Leere Zellen werden bei der Berechnung des Durchschnitts ignoriert.

**Anmerkung:** MEAN ist mit der Makrofunktion AVG identisch.

## Beispiele

```
TEMP = MEAN(V1)
```

Erstellt eine neue Spalte TEMP, die einen Einzelwert enthält, der das arithmetische Mittel des Inhalts von Spalte V1 darstellt.

```
TEMP = MEAN(V1:V3)
```

Erstellt eine neue Spalte TEMP, die einen Einzelwert enthält, der das arithmetische Mittel des Inhalts der Spalten V1, V2 und V3 darstellt.

TEMP = MEAN(V1[10:20])
Erstellt eine neue Spalte TEMP, die einen Einzelwert enthält, der das arithmetische Mittel der Zellen in Zeile 10-20 von Spalte V1 darstellt.
TEMP = MEAN(V1[1:5]:V4)
Erstellt eine neue Spalte TEMP, die einen Einzelwert enthält, der das arithmetische Mittel der Zellen in Zeile 1-5 von Spalte V1 bis V4 darstellt.
TEMP = MEAN(V1:V3, COL)
Erstellt drei neue Spalten TEMP, VX und VY. Der Einzelwert in der Spalte TEMP ist das arithmetische Mittel des Inhalts von Spalte V1, der Einzelwert in der Spalte VX ist das arithmetische Mittel des Inhalts von Spalte V2 und der Einzelwert in der Spalte VY ist das arithmetische Mittel des Inhalts von Spalte V3.
TEMP = MEAN(V1[10:20]:V3, COL)
Erstellt drei neue Spalten TEMP, VX und VY, die jeweils einen Einzelwert enthalten. Der Wert in der Spalte TEMP ist das arithmetische Mittel der Zellen in Zeile 10-20 von Spalte V1, der Wert in der Spalte VX ist das arithmetische Mittel der Zellen in Zeile 10-20 von Spalte V2 und der Wert in der Spalte VY ist das arithmetische Mittel der Zellen in Zeile 10-20 von Spalte V3.
TEMP = MEAN(V1:V3, ROW)
Erstellt eine neue Spalte TEMP, in der jeder Zelleneintrag das arithmetische Mittel der entsprechenden Zeile in Spalte V1, V2 und V3 darstellt.
TEMP = MEAN(V1[1:5]:V3, ROW)
Erstellt eine neue Spalte TEMP, in der die Zellen in Zeile 1-5 das arithmetische Mittel der entsprechenden Zeile in Spalte V1 bis V3 enthalten. Die anderen Zellen in TEMP sind leer.

## Zugehörige Funktionen

Funktion	Beschreibung
SUM oder TOTAL	Berechnet die Summe eines Zellenbereichs.

## MIN-Makro

Das MIN-Makro ist in IBM Campaign und IBM Interact verfügbar.

### Syntax

MIN(data [, keyword])

### Parameter

data

Die numerischen Werte, deren Minimum berechnet werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

keyword

Dieses optionale Schlüsselwort legt fest, wie die Berechnung auf den Eingabedatenbereich angewendet wird. Wählen Sie eine der folgenden Angaben aus:

ALL - Wendet die Berechnung auf alle Zellen in data an (Standard)

COL - Führt die Berechnung für jede Spalte von data gesondert aus

ROW - Führt die Berechnung für jede Zeile von data gesondert aus

Weitere Informationen zur Verwendung von Schlüsselwörtern finden Sie unter „DATE“ auf Seite 52.

**Anmerkung:** Viele Makrofunktionen setzen die Schlüsselwortparameter {ALL | COL | ROW} ein. Diese Schlüsselwörter gelten nicht für **IBM Campaign**, da es sich bei den Eingabedaten immer um eine Einzelspalte oder ein Einzelfeld handelt. Das Makro verhält sich immer so, als ob das Schlüsselwort COL angegeben würde. Daher brauchen Sie diese Schlüsselwörter nicht anzugeben, wenn Sie **IBM Campaign** verwenden.

## Beschreibung

MIN berechnet das Minimum aller Zellen im angegebenen Datenbereich. Es wird eine einzelne neue Spalte zurückgegeben, die den Minimalwert enthält.

## Beispiele

TEMP = MIN(V1)	Erstellt eine neue Spalte TEMP, die einen Einzelwert enthält, der den Minimalwert des Inhalts von Spalte V1 darstellt.
TEMP = MIN(V1:V3)	Erstellt eine neue Spalte TEMP, die einen Einzelwert enthält, der das Minimum des Inhalts der Spalten V1, V2 und V3 darstellt.
TEMP = MIN(V1[10:20])	Erstellt eine neue Spalte TEMP, die einen Einzelwert enthält, der das Minimum der Zellen in Zeile 10-20 von Spalte V1 darstellt.
TEMP = MIN(V1[1:5]:V4)	Erstellt eine neue Spalte TEMP, die einen Einzelwert enthält, der das Minimum der Zellen in Zeile 1-5 von Spalte V1 bis V4 darstellt.
TEMP = MIN(V1:V3, COL)	Erstellt drei neue Spalten TEMP, VX und VY. Der Einzelwert in der Spalte TEMP ist das Minimum des Inhalts von Spalte V1, der Einzelwert in der Spalte VX ist das Minimum des Inhalts von Spalte V2 und der Einzelwert in der Spalte VY ist das Minimum des Inhalts von Spalte V3.
TEMP = MIN(V1[1:5]:V3, COL)	Erstellt drei neue Spalten TEMP, VX und VY, die jeweils einen Einzelwert enthalten. Der Wert in der Spalte TEMP ist das Minimum der Zellen in Zeile 1-5 von Spalte V1, der Wert in der Spalte VX ist das Minimum der Zellen in Zeile 1-5 von Spalte V2 und der Wert in der Spalte VY ist das Minimum der Zellen in Zeile 1-5 von Spalte V3.

TEMP = MIN(V1:V3, ROW)
Erstellt eine neue Spalte TEMP, in der jeder Zelleneintrag das Minimum der entsprechenden Zeile in Spalte V1, V2 und V3 darstellt.
TEMP = MIN(V1[10:20]:V3, ROW)
Erstellt eine neue Spalte TEMP, in der die ersten 11 Zellen das Minimum der Werte in Zeile 1-5 von Spalte V1 bis V3 enthalten. Die anderen Zellen in TEMP sind leer.

## Zugehörige Funktionen

Funktion	Beschreibung
MAX	Berechnet das Maximum eines Zellenbereichs
MAX_TO_INDEX	Gibt für jede Zeile der angegebenen Spalte den Spaltenindex des Maximalwerts zurück.

---

## MINUS-Makro

Das MINUS-Makro Makro ist in IBM Campaign und IBM Interact verfügbar.

### Syntax

data MINUS subtrahend data - subtrahend

### Parameter

data

Der Zellenbereich mit Zahlen, von denen etwas subtrahiert werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

subtrahend

Die Zahl(en), die von allen Werten in der angegebenen Spalte subtrahiert werden soll(en). Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Die Anzahl der Spalten in subtrahend muss mit der Anzahl der Spalten in data übereinstimmen, es sei denn, bei subtrahend handelt es sich um eine Konstante. Informationen zur Formatdefinition von subtrahend (identisch mit data) finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

### Beschreibung

MINUS subtrahiert subtrahend vom angegebenen Datenbereich data. MINUS gibt für jede Eingabespalte eine neue Spalte zurück, wobei jeweils die entsprechende Spalte von subtrahend von der entsprechenden Spalte von data subtrahiert wird (d. h., von der ersten Spalte von data wird die erste Spalte von subtrahend subtrahiert, von der zweiten Spalte die zweite Spalte usw.).

Wenn es sich bei subtrahend um eine Konstante handelt, wird der betreffende Wert von jedem Wert in data subtrahiert. Wenn subtrahend eine oder mehrere Spalten enthält, werden die Berechnungen zeilenweise zwischen einer Spalte von data und einer Spalte von subtrahend ausgeführt. Von der ersten Zeile von data wird der erste Zeilenwert von subtrahend subtrahiert, von der zweiten Zeile der zweite Zeilenwert usw. Diese zeilenweise ausgeführte Berechnung erzeugt für jede Zeile ein Ergebnis, bis zum letzten Wert der kürzesten Spalte.

**Anmerkung:** Der Operator MINUS kann durch ein Minuszeichen oder einen Bindestrich (-) abgekürzt werden.

## Beispiele

TEMP = 7 MINUS 4 oder TEMP = 7 - 4
Erstellt eine neue Spalte TEMP, die den Wert drei enthält.
TEMP = V1 - 8
Erstellt eine neue Spalte TEMP, in der jeder Wert den Inhalt von Spalte V1 minus acht darstellt.
TEMP = V1:V3 - 2
Erstellt drei neue Spalten TEMP, VX und VY. Die Werte in der Spalte TEMP sind der Inhalt von Spalte V1 minus zwei, die Werte von Spalte VX sind der Inhalt von Spalte V2 minus zwei und die Werte von Spalte VY sind der Inhalt von Spalte V3 minus zwei.
TEMP = V1 - V1
Erstellt eine neue Spalte TEMP, die nur Nullen enthält (jede Zahl minus dieselbe Zahl ergibt null).
TEMP = V1 - V2
Erstellt eine neue Spalte TEMP, wobei jeder Wert den Zeilenwert von Spalte V1 minus den entsprechenden Zeilenwert von Spalte V2 darstellt.
TEMP = V1:V3 -V4:V6
Erstellt drei neue Spalten TEMP, VX und VY. Die Spalte TEMP enthält die Werte von Spalte V1 minus die entsprechenden Zeilenwerte von Spalte V4. Die Spalte VX subtrahiert Spalte V5 von Spalte V2. Die Spalte VY subtrahiert Spalte V6 von Spalte V3.
TEMP = V1[10:20] - V2 oder TEMP = V1[10:20] - V2[1:11]
Erstellt eine neue Spalte TEMP, deren erste 11 Zellen die Werte in Zeile 10-20 von Spalte V1 minus die Werte in Zeile 1-11 von Spalte V2 enthalten. Die anderen Zellen in TEMP sind leer.

## Zugehörige Funktionen

Funktion	Beschreibung
PLUS	Addiert die Inhalte zweier Datenbereiche.
SUM oder TOTAL	Berechnet die Summe eines Zellenbereichs.

## MOD-Makro

Das MOD-Makro Makro ist in IBM Campaign und IBM Interact verfügbar.

## Syntax

`data MOD divisor data % divisor`

## Parameter

`data`

Die Ganzzahlwerte, deren Modulo-Wert berechnet werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

`divisor`

Die Ganzzahl ungleich null, bezüglich deren der Modulo-Wert berechnet werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Die Anzahl der Spalten in `divisor` muss mit der Anzahl der Spalten in `data` übereinstimmen, es sei denn, bei `divisor` handelt es sich um eine Konstante. Informationen zur Formatdefinition von `divisor` (identisch mit `data`) finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

## Beschreibung

MOD berechnet den Rest der Division des angegebenen Datenbereichs durch einen angegebenen Wert. Dieser wird berechnet, indem jeder Wert durch `divisor` geteilt und der Rest zurückgegeben wird. MOD gibt für jede Eingabespalte eine neue Spalte zurück, die die Zahlen in `data` modulo `divisor` enthält. Der Rest hat dasselbe Vorzeichen (positiv oder negativ) wie `data`.

Wenn es sich bei `divisor` um eine Konstante handelt, wird für jeden Wert in der angegebenen Spalte der betreffende Modulo-Wert berechnet. Wenn es sich bei `divisor` um eine Spalte handelt, werden die Berechnungen zeilenweise ausgeführt. Bei der Modulo-Berechnung wird die erste Zeile in `data` dem ersten Zeilenwert von `divisor` zugeordnet, die zweite Zeile dem zweiten Zeilenwert usw. Diese zeilenweise ausgeführte Berechnung erzeugt für jede Zeile ein Ergebnis, bis zum letzten Wert der kürzesten Spalte.

**Anmerkung:** Wenn `divisor` null ist, wird der Fehler 'Division durch null' zurückgegeben.

**Anmerkung:** Der Operator MOD kann durch ein Prozentzeichen (%) abgekürzt werden. Beispielsweise ist `TEMP = 5 % 3` äquivalent zu `TEMP = 5 MOD 3`.

## Beispiele

<code>TEMP = 10 MOD 8</code> oder <code>TEMP = 10 % 8</code>
Erstellt eine neue Spalte TEMP, die den Wert 2 enthält.
<code>TEMP = -10 % 8</code>
Erstellt eine neue Spalte TEMP, die den Wert -2 enthält.

TEMP = V1 % 8
Erstellt eine neue Spalte TEMP, in der jeder Wert den Inhalt von Spalte V1 modulo acht darstellt.
TEMP = V1:V3 % 2
Erstellt drei neue Spalten TEMP, VX und VY. Die Werte in der Spalte TEMP sind die Werte modulo zwei des Inhalts von Spalte V1, die Werte von Spalte VX sind die Werte modulo zwei des Inhalts von Spalte V2 und die Werte von Spalte VY sind die Werte modulo zwei des Inhalts von Spalte V3.
TEMP = V1 % V1
Erstellt eine neue Spalte TEMP, die für jeden Eintrag in Spalte V1 eine Null enthält. Jede Zahl modulo die Zahl selbst ergibt null.
TEMP = V1 % V2
Erstellt eine neue Spalte TEMP, wobei jeder Wert den Zeilenwert von Spalte V1 modulo den entsprechenden Zeilenwert von Spalte V2 darstellt. Wenn V2=V1, werden nur Nullen zurückgegeben, wie im vorherigen Beispiel.
TEMP = V1:V3 % V4:V6
Erstellt drei neue Spalten TEMP, VX und VY. Die Spalte TEMP enthält die Werte von Spalte V1 modulo die entsprechenden Zeilenwerte von Spalte V4. Die Spalte VX enthält die Ergebnisse von Spalte V2 modulo V5. Die Spalte VY enthält die Ergebnisse von Spalte V3 modulo V6.
TEMP = V1[10:20] % V2 oder TEMP = V1[10:20] % V2[1:11]
Erstellt eine neue Spalte TEMP, deren erste 11 Zellen die Werte in Zeile 10-20 von Spalte V1 modulo die Werte in Zeile 1-11 von Spalte V2 darstellen. Die anderen Zellen in TEMP sind leer.

## Zugehörige Funktionen

Funktion	Beschreibung
DIV	Dividiert einen angegebenen Datenbereich durch einen anderen.
MOD	Berechnet den Modulo-Wert des Inhalts des angegebenen Datenbereichs.

## MONTHOF-Makro

Das MONTHOF-Makro Makro ist nur in IBM Campaign verfügbar.

### Syntax

MONTHOF(date\_string [, input\_format])

### Parameter

date\_string

Ein Text, der ein gültiges Datum darstellt.

input\_format

Eines der Schlüsselwörter in der folgenden Tabelle, das das Datumsformat von `date_string` angibt.

## Beschreibung

`MONTHOF` gibt für das durch `date_string` angegebene Datum den Monat als Zahl zurück. Wenn `input_format` nicht angegeben ist, wird das Standardschlüsselwort `DELIM_M_D_Y` verwendet.

## Beispiele

`MONTHOF("012171", MMDDYY)` gibt die Zahl 1 zurück.

**Anmerkung:** Weitere Informationen zu gültigen Datumsformaten finden Sie unter „DATE“ auf Seite 52.

## Zugehörige Funktionen

Funktion	Beschreibung
<code>DAYOF</code>	Gibt den Wochentag als Zahl zurück.
<code>WEEKDAYOF</code>	Gibt den Wochentag der Woche als Zahl zurück.
<code>YEAROF</code>	Gibt das Jahr als Zahl zurück.

---

## MULT-Makro

Das `MULT`-Makro Makro ist in IBM Campaign und IBM Interact verfügbar.

### Syntax

```
data MULT multiplier data * multiplier
```

### Parameter

`data`

Die numerischen Werte, die multipliziert werden sollen. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

`multiplier`

Die Zahl, mit der alle Werte in der angegebenen Spalte multipliziert werden sollen. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Die Anzahl der Spalten in `multiplier` muss mit der Anzahl der Spalten in `data` übereinstimmen, es sei denn, bei `multiplier` handelt es sich um eine Konstante. Informationen zur Formatdefinition von `multiplier` (identisch mit `data`) finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

## Beschreibung

MULT multipliziert die Werte in den beiden angegebenen Datenbereichen. MULT gibt für jede Eingabespalte eine neue Spalte zurück, die die mit `multiplier` multiplizierten Zahlen in `data` enthält. Wenn es sich bei `multiplier` um eine Konstante handelt, wird jeder Wert in `data` mit dem betreffenden Wert multipliziert. Wenn es sich bei `multiplier` um eine Spalte handelt, werden die Berechnungen zeilenweise ausgeführt. Die erste Zeile von `data` wird mit der ersten Zeile von `multiplier` multipliziert, die zweite Zeile mit der zweiten Zeile usw. Diese zeilenweise ausgeführte Berechnung erzeugt für jede Zeile ein Ergebnis, bis zum letzten Wert der kürzesten Spalte.

**Anmerkung:** Wenn eine Spalte in jeder Zeile dieselbe Zahl  $x$  wie `multiplier` enthält, ist dies dasselbe, als wenn als `multiplier` die Konstante  $x$  verwendet wird.

**Anmerkung:** Der Operator MULT kann durch einen Stern (\*) abgekürzt werden.

## Beispiele

TEMP = 8 MULT 4 oder TEMP = 8 * 4
Erstellt eine neue Spalte TEMP, die den Wert 32 enthält.
TEMP = V1 * 8
Erstellt eine neue Spalte TEMP, in der jeder Wert den Inhalt von Spalte V1 darstellt, der mit acht multipliziert ist.
TEMP = V1:V3 * 2
Erstellt drei neue Spalten TEMP, VX und VY. Die Werte in der Spalte TEMP sind jeweils das Zweifache des Inhalts von Spalte V1, die Werte von Spalte VX sind das Zweifache des Inhalts von Spalte V2 und die Werte von Spalte VY sind das Zweifache des Inhalts von Spalte V3.
TEMP = V1 * V1
Erstellt eine neue Spalte TEMP, die das Quadrat jedes Werts in Spalte V1 enthält.
TEMP = V1 * V2
Erstellt eine neue Spalte TEMP, wobei jeder Zeilenwert von Spalte V1 mit dem entsprechenden Zeilenwert von Spalte V2 multipliziert ist.
TEMP = V1:V3 * V4:V6
Erstellt drei neue Spalten TEMP, VX und VY. In der Spalte TEMP sind die Werte von Spalte V1 mit den entsprechenden Zeilenwerten von Spalte V4 multipliziert. Die Spalte VX enthält die Multiplikation von Spalte V2 mit Spalte V5. Die Spalte VY enthält die Multiplikation von Spalte V3 mit Spalte V6.
TEMP = V1[10:20] * V2 oder TEMP = V1[10:20] * V2[1:11]
Erstellt eine neue Spalte TEMP, deren erste 11 Zellen das Ergebnis der Multiplikation der Werte in Zeile 10-20 von Spalte V1 mit den Werten in Zeile 1-11 von Spalte V2 enthalten. Die anderen Zellen in TEMP sind leer.

## Zugehörige Funktionen

Funktion	Beschreibung
DIV	Dividiert einen angegebenen Datenbereich durch einen anderen.

Funktion	Beschreibung
EXP	Erhebt die natürliche Zahl (e) in die Potenz, die durch den Inhalt jeder Zelle im angegebenen Datenbereich angegeben ist.
POW	Erhebt einen Basiswert in die Potenz des oder der angegebenen Exponenten.

---

## NE-Makro

Das NE-Makro Makro ist in IBM Campaign und IBM Interact verfügbar.

### Syntax

```
data1 NE data2 data1 != data2 data1 <> data2
```

### Parameter

data1

Der Zellenbereich, der verglichen werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

data2

Die Zahlen, mit denen alle Werte in der angegebenen Spalte verglichen werden sollen. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Die Anzahl der Spalten in data2 muss mit der Anzahl der Spalten in data1 übereinstimmen, es sei denn, bei data2 handelt es sich um eine Konstante. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

### Beschreibung

NE vergleicht die beiden angegebenen Datenbereiche und gibt eine Eins zurück, wenn die Werte ungleich sind, bzw. eine Null, wenn die Werte gleich sind. NE gibt für jede Eingabespalte eine neue Spalte zurück, wobei jeweils die entsprechende Spalte von data1 mit der entsprechenden Spalte von data2 verglichen wird (d. h., die erste Spalte von data1 wird mit der ersten Spalte von data2 verglichen, die zweite Spalte mit der zweiten Spalte usw.).

Wenn es sich bei data2 um eine Konstante handelt, wird jeder Wert in data1 mit dem betreffenden Wert verglichen. Wenn es sich bei data2 um eine Spalte handelt, werden die Berechnungen zeilenweise ausgeführt. Die erste Zeile von data1 wird mit der ersten Zeile von data2 verglichen, die zweite Zeile mit der zweiten Zeile usw. Diese zeilenweise ausgeführte Berechnung erzeugt für jede Zeile ein Ergebnis, bis zum letzten Wert der kürzesten Spalte.

**Anmerkung:** Wenn eine Spalte in jeder Zeile dieselbe Zahl  $x$  wie data2 enthält, ist dies dasselbe, als wenn als data2 die Konstante  $x$  verwendet wird.

**Anmerkung:** Der Operator NE kann durch die Folge aus Ausrufezeichen und Gleichheitszeichen (!=) oder durch die Folge aus Kleiner-als-Zeichen und Größer-als-Zeichen (<>) abgekürzt werden.

## Beispiele

TEMP = 3 NE 4 oder TEMP = 3 != 4; TEMP = 3 <> 4
Erstellt eine neue Spalte TEMP, die den Wert eins enthält (drei ist ungleich vier).
TEMP = V1 != 8
Erstellt eine neue Spalte TEMP, in der jeder Wert eins ist, wenn der entsprechende Zeilenwert von Spalte V1 ungleich der Zahl acht ist; andernfalls ist der Wert null.
TEMP = V1:V3 != 2
Erstellt drei neue Spalten TEMP, VX und VY. Die Werte in der Spalte TEMP sind der mit zwei verglichene Inhalt von Spalte V1, die Werte von Spalte VX sind der mit zwei verglichene Inhalt von Spalte V2 und die Werte von Spalte VY sind der mit zwei verglichene Inhalt von Spalte V3.
TEMP = V1 != V1
Erstellt eine neue Spalte TEMP, die nur Nullen enthält (jede Zahl ist gleich sich selbst).
TEMP = V1 != V2
Erstellt eine neue Spalte TEMP, wobei jeder Zeilenwert von Spalte V1 mit dem entsprechenden Zeilenwert von Spalte V2 verglichen wird.
TEMP = V1:V3 != V4:V6
Erstellt drei neue Spalten TEMP, VX und VY. In der Spalte TEMP sind die Werte von Spalte V1 mit den entsprechenden Zeilenwerten von Spalte V4 verglichen. Die Spalte VX enthält den Vergleich von Spalte V2 mit Spalte V5. Die Spalte VY enthält den Vergleich von Spalte V3 mit Spalte V6.
TEMP = V1[10:20] != V2 oder TEMP = V1[10:20] != V2[1:11]
Erstellt eine neue Spalte TEMP, deren erste 11 Zellen das Ergebnis des Vergleichs der Werte in Zeile 10-20 von Spalte V1 mit den Werten in Zeile 1-11 von Spalte V2 enthalten. Die anderen Zellen in TEMP sind leer.

## Zugehörige Funktionen

Funktion	Beschreibung
EQ	Gibt TRUE zurück, wenn der eine Datenbereich dem anderen gleich ist.
GE	Gibt TRUE zurück, wenn der eine Datenbereich größer-gleich dem anderen ist.
GT	Gibt TRUE zurück, wenn der eine Datenbereich größer als der andere ist.
LE	Gibt TRUE zurück, wenn der eine Datenbereich kleiner-gleich dem anderen ist.
LT	Gibt TRUE zurück, wenn der eine Datenbereich kleiner als der andere ist.

---

## NOT-Makro

Das NOT-Makro Makro ist in IBM Campaign und IBM Interact verfügbar.

### Syntax

NOT(data) ! data

### Parameter

data

Die numerischen Werte, deren logisches Nicht berechnet werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

### Beschreibung

NOT gibt das logische Nicht der Werte im angegebenen Datenbereich zurück. NOT gibt für jede Eingabespalte eine neue Spalte zurück, die das logische Nicht der Werte in der entsprechenden Eingabespalte enthält. Diese Funktion gibt bei Werten ungleich null eine Null und bei Werten gleich null eine Eins zurück.

**Anmerkung:** Der Operator NOT kann durch ein Ausrufezeichen (!) abgekürzt werden. Stellen Sie das Ausrufezeichen dem Datenwert voran (statt z. B. NOT(V1) anzugeben, können Sie einfach !V1 eingeben).

### Beispiele

TEMP = NOT(3.2) oder TEMP = !1
Erstellt eine neue Spalte TEMP, die den Wert null enthält.
TEMP = !0 oder TEMP = !(2+2=3)
Erstellt eine neue Spalte TEMP, die den Wert eins enthält.
TEMP = !V1
Erstellt eine neue Spalte TEMP, in der jeder Wert das logische Nicht der Werte in Spalte V1 darstellt.
TEMP = !V1:V3
Erstellt drei neue Spalten TEMP, VX und VY. Die Werte in der Spalte TEMP sind jeweils das logische Nicht der Werte in Spalte V1, die Werte von Spalte VX sind das logische Nicht der Werte in Spalte V2 und die Werte von Spalte VY sind das logische Nicht der Werte in Spalte V3.
TEMP = !V1[10:20]
Erstellt eine neue Spalte TEMP, in der die ersten 11 Zellen jeweils das logische Nicht der Werte in Zeile 10-20 von Spalte V1 darstellen. Die anderen Zellen in TEMP sind leer.
TEMP = !V1[1:5]:V2
Erstellt zwei neue Spalten TEMP und VX, die jeweils die Werte in Zeile 1-5 enthalten (die anderen Zellen sind leer). Die Werte in der Spalte TEMP sind das logische Nicht der Werte der entsprechenden Zeilen von Spalte V1, die Werte in der Spalte VX sind das logische Nicht der Werte der entsprechenden Zeilen von Spalte V2.

## Zugehörige Funktionen

Funktion	Beschreibung
UND	Berechnet das logische Und zwischen den beiden angegebenen Datenbereichen.
INVERSE	Berechnet den Negativwert des Inhalts des angegebenen Datenbereichs.
ODER	Berechnet das logische Oder zwischen den beiden angegebenen Datenbereichen.
SIGN	Berechnet das Vorzeichen (positiv oder negativ) der Werte im angegebenen Datenbereich.

---

## NUMBER-Makro

Das NUMBER-Makro ist in IBM Campaign und IBM Interact verfügbar.

### Syntax

NUMBER(data [, conversion\_keyword])

### Parameter

data

Die ASCII-Textdaten, die in numerische Werte konvertiert werden sollen. Dabei kann es sich um ASCII-Text in Anführungszeichen, eine Textspalte, einen Zellenbereich mit Text oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

conversion\_keyword

Dieses optionale Schlüsselwort legt fest, wie Textformate für Datums- und Uhrzeitangaben interpretiert werden sollen. Wählen Sie eines der Schlüsselwörter in der folgenden Tabelle aus.

**Anmerkung:** Wird dieser Parameter nicht angegeben, ist der Standard 1.

Schlüsselwort	Format	Beschreibung
0	#####	Konvertiert die ersten 5 Zeichen jeder Zeichenfolge in eine eindeutige Zahl
1	\$ ( default)	Konvertiert Dollarwerte in numerische Werte (z. B. "\$123.45" in 123.45)
2	%	Konvertiert Prozentwerte in numerische Werte (z. B. "50%" in 0.5)
3	mm/dd/yy hh:mm	Konvertiert ein Datum und eine Uhrzeit in die Anzahl der Tage, die seit dem 1. Januar 0000 vergangen sind (zum Jahr yy wird automatisch 1900 addiert)

Schlüsselwort	Format	Beschreibung
4	dd-mmm-yy	Konvertiert ein Datum in die Anzahl der Tage, die seit dem 1. Januar 0000 vergangen sind (zum Jahr yy wird automatisch 1900 addiert)
5	mm/dd/yy	Konvertiert ein Datum in die Anzahl der Tage, die seit dem 1. Januar 0000 vergangen sind (zum Jahr yy wird automatisch 1900 addiert)
6	mmm-yy	Konvertiert ein Datum in die Anzahl der Tage, die zwischen dem ersten Tag des angegebenen Monats und dem 1. Januar 0000 vergangen sind (zum Jahr yy wird automatisch 1900 addiert)
7	dd-mmm	Konvertiert ein Datum in die Anzahl der Tage, die seit dem Anfang des Jahres vergangen sind (z. B. "01-FEB" in 32)
8	mmm	Konvertiert eine Monatsabkürzung aus 3 Buchstaben in einen Wert zwischen 1 und 12 (z. B. "DEC" in 12)
9	{January   February   March ... }	Konvertiert einen vollständigen Monatsnamen in einen Wert zwischen 1 und 12 (z. B. "March" in 3)
10	{Sun   Mon   Tue ... }	Konvertiert eine Wochentagsabkürzung aus 3 Buchstaben in einen Wert zwischen 0 und 6, wobei die Woche am Sonntag beginnt (z. B. "Sun" in 0)
11	{Sunday   Monday   Tuesday ... }	Konvertiert einen vollständigen Wochentagsnamen in einen Wert zwischen 0 und 6, wobei die Woche am Sonntag beginnt (z. B. "Monday" in 1)
12	hh:mm:ss {AM   PM}	Konvertiert die Uhrzeit in die Anzahl der Sekunden, die seit Mitternacht vergangen sind (z. B. "01:00:00 AM" in 3600)
13	hh:mm:ss	Konvertiert die Uhrzeit in die Anzahl der Sekunden, die seit Mitternacht vergangen sind (z. B. "01:00:00" in 3600)
14	hh:mm {AM   PM}	Konvertiert die Uhrzeit in die Anzahl der Minuten, die seit Mitternacht vergangen sind (z. B. "01:00 AM" in 60)
15	hh:mm	Konvertiert die Uhrzeit in die Anzahl der Minuten, die seit Mitternacht vergangen sind (z. B. "01:00" in 60)
16	mm:ss	Konvertiert die Uhrzeit in die Anzahl der Sekunden, die seit Mitternacht vergangen sind (z. B. "30:00" in 1800)
17	ddmm	Konvertiert ein Datum in die Anzahl der Tage, die seit dem Anfang des Jahres vergangen sind (z. B. "3101" in 31)
18	ddmmm	Konvertiert ein Datum in die Anzahl der Tage, die seit dem Anfang des Jahres vergangen sind (z. B. "31JAN" in 31)

Schlüsselwort	Format	Beschreibung
19	ddmmyy	Konvertiert ein Datum in die Anzahl der Tage, die seit dem 1. Januar 0000 vergangen sind (wenn das Jahr yy kleiner-gleich 20 ist, wird automatisch 1900 addiert; andernfalls wird 2000 addiert)
20	ddmmyyyy	Konvertiert ein Datum in die Anzahl der Tage, die seit dem 1. Januar 0000 vergangen sind (z. B. "31JAN0000" in 31)
21	ddmmyy	Konvertiert ein Datum in die Anzahl der Tage, die seit dem 1. Januar 0000 vergangen sind (wenn das Jahr yy kleiner-gleich 20 ist, wird automatisch 1900 addiert; andernfalls wird 2000 addiert)
22	ddmmyyyy	Konvertiert ein Datum in die Anzahl der Tage, die seit dem 1. Januar 0000 vergangen sind (z. B. "31010000" in 31)
23	mmdd	Konvertiert ein Datum in die Anzahl der Tage, die seit dem Anfang des Jahres vergangen sind (z. B. "0131" in 31)
24	mmddy	Konvertiert ein Datum in die Anzahl der Tage, die seit dem 1. Januar 0000 vergangen sind (wenn das Jahr yy kleiner-gleich 20 ist, wird automatisch 1900 addiert; andernfalls wird 2000 addiert)
25	mmddyyyy	Konvertiert ein Datum in die Anzahl der Tage, die seit dem 1. Januar 0000 vergangen sind (z. B. "01010001" in 366)
26	mmm	Konvertiert eine Monatsabkürzung aus 3 Buchstaben in einen Wert zwischen 1 und 12 (z. B. "MAR" in 3) [identisch mit Schlüsselwort 8]
27	mmdd	Konvertiert ein Datum in die Anzahl der Tage, die seit dem Anfang des Jahres vergangen sind (z. B. "JAN31" in 31)
28	mmmdyy	Konvertiert ein Datum in die Anzahl der Tage, die seit dem 1. Januar 0000 vergangen sind (wenn das Jahr yy kleiner-gleich 20 ist, wird automatisch 1900 addiert; andernfalls wird 2000 addiert)
29	mmmdyyyy	Konvertiert ein Datum in die Anzahl der Tage, die seit dem 1. Januar 0000 vergangen sind (z. B. "FEB010000" in 32)
30	mmyy	Konvertiert ein Datum in die Anzahl der Tage, die zwischen dem ersten Tag des angegebenen Monats und dem 1. Januar 0000 vergangen sind (wenn das Jahr yy kleiner-gleich 20 ist, wird automatisch 1900 addiert; andernfalls wird 2000 addiert)
31	mmmyyyy	Konvertiert ein Datum in die Anzahl der Tage, die zwischen dem ersten Tag des angegebenen Monats und dem 1. Januar 0000 vergangen sind (z. B. "FEB010000" in 32)

<b>Schlüsselwort</b>	<b>Format</b>	<b>Beschreibung</b>
32	myyy	Konvertiert ein Datum in die Anzahl der Tage, die zwischen dem ersten Tag des angegebenen Monats und dem 1. Januar 0000 vergangen sind (wenn das Jahr yy kleiner-gleich 20 ist, wird automatisch 1900 addiert; andernfalls wird 2000 addiert)
33	myyyyy	Konvertiert ein Datum in die Anzahl der Tage, die zwischen dem ersten Tag des angegebenen Monats und dem 1. Januar 0000 vergangen sind (z. B. "020000" in 32)
34	ymmm	Konvertiert ein Datum in die Anzahl der Tage, die zwischen dem ersten Tag des angegebenen Monats und dem 1. Januar 0000 vergangen sind (wenn das Jahr yy kleiner-gleich 20 ist, wird automatisch 1900 addiert; andernfalls wird 2000 addiert)
35	ymddd	Konvertiert ein Datum in die Anzahl der Tage, die seit dem 1. Januar 0000 vergangen sind (wenn das Jahr yy kleiner-gleich 20 ist, wird automatisch 1900 addiert; andernfalls wird 2000 addiert)
36	ymmm	Konvertiert ein Datum in die Anzahl der Tage, die zwischen dem ersten Tag des angegebenen Monats und dem 1. Januar 0000 vergangen sind (wenn das Jahr yy kleiner-gleich 20 ist, wird automatisch 1900 addiert; andernfalls wird 2000 addiert)
37	ymmmddd	Konvertiert ein Datum in die Anzahl der Tage, die seit dem 1. Januar 0000 vergangen sind (wenn das Jahr yy kleiner-gleich 20 ist, wird automatisch 1900 addiert; andernfalls wird 2000 addiert)
38	yyyy	Konvertiert das Jahr in die Anzahl der Jahre, die seit dem Jahr 0000 vergangen sind (z. B. "1998" in 1998)
39	yyyyymm	Konvertiert ein Datum in die Anzahl der Tage, die zwischen dem ersten Tag des angegebenen Monats und dem 1. Januar 0000 vergangen sind (z. B. "000102" in 32)
40	yyyymddd	Konvertiert ein Datum in die Anzahl der Tage, die seit dem 1. Januar 0000 vergangen sind (z. B. "00000201" in 32)
41	yyyymmm	Konvertiert ein Datum in die Anzahl der Tage, die zwischen dem ersten Tag des angegebenen Monats und dem 1. Januar 0000 vergangen sind (z. B. "000102" in 32)
42	yyyymmddd	Konvertiert ein Datum in die Anzahl der Tage, die seit dem 1. Januar 0000 vergangen sind (z. B. "0000FEB01" in 32)
43	<day>* <month>	Konvertiert ein Datum mit Begrenzer, das aus Tag und Monat besteht, in die Anzahl der Tage, die seit dem Anfang des Jahres vergangen sind (z. B. "15-JAN" in 15)

Schlüsselwort	Format	Beschreibung
44	<day>* <month>* <year>	Konvertiert ein Datum mit Begrenzer, das aus Tag, Monat und Jahr besteht, in die Anzahl der Tage, die seit dem 1. Januar 0000 vergangen sind (z. B. "1/1/0001" in 366)
45	<month>* <day>	Konvertiert ein Datum mit Begrenzer, das aus Monat und Tag besteht, in die Anzahl der Tage, die seit dem Anfang des Jahres vergangen sind (z. B. "JAN 31" in 31)
46	<month>* <day>* <year>	Konvertiert ein Datum mit Begrenzer, das aus Monat, Tag und Jahr besteht, in die Anzahl der Tage, die seit dem 1. Januar 0000 vergangen sind (z. B. "JAN 1, 0001" in 366)
47	<month>* <year>	Konvertiert ein Datum mit Begrenzer, das aus Monat und Jahr besteht, in die Anzahl der Tage, die zwischen dem ersten Tag des angegebenen Monats und dem 1. Januar 0000 vergangen sind
48	<year>* <month>	Konvertiert ein Datum mit Begrenzer, das aus Jahr und Monat besteht, in die Anzahl der Tage, die zwischen dem ersten Tag des angegebenen Monats und dem 1. Januar 0000 vergangen sind
49	<year>* <month>* <day>	Konvertiert ein Datum mit Begrenzer, das aus Jahr, Monat und Tag besteht, in die Anzahl der Tage, die seit dem 1. Januar 0000 vergangen sind (z. B. "0001/01/01" in 366)
50	yy	Konvertiert das Jahr in die Anzahl der Jahre, die seit dem Jahr 0000 vergangen sind (z. B. "97" in 97)
51	mm	Konvertiert den Monat in einen Wert zwischen 1 und 12 (z. B. "SEP" in 9)
52	dd	Konvertiert den Tag in einen Wert zwischen 1 und 31 (z. B. "28" in 28)
53	{January   February   March ... }	Konvertiert einen vollständigen Monatsnamen in einen Wert zwischen 1 und 12 (z. B. "March" in 3) [identisch mit Schlüsselwort 9]
54	{Sunday   Monday   Tuesday ... }	Konvertiert einen vollständigen Wochentagsnamen in einen Wert zwischen 1 und 7, wobei die Woche am Sonntag beginnt (z. B. "Sunday" in 1)
55	{Sun   Mon   Tue ... }	Konvertiert eine Wochentagsabkürzung aus 3 Buchstaben in einen Wert zwischen 1 und 7, wobei die Woche am Sonntag beginnt (z. B. "Sun" in 1)

## Beschreibung

NUMBER konvertiert Textwerte im angegebenen Datenbereich anhand des angegebenen Umwandlungsformats für Datums- und Uhrzeitangaben in numerische Werte. Wenn eine Zeichenfolge mit dem angegebenen `conversion_keyword` nicht geparkt werden kann, gibt NUMBER einen Fehler zurück. Format 0 konvertiert die ersten fünf Zeichen jeder Zeichenfolge für jede eindeutige Zeichenfolge in eine unterschiedli-

che Zahl. Dies ist ein einfacher Weg, eine Textspalte in eindeutige Klassen umzuwandeln, die als Ausgaben für einen Klassifikator dienen können.

Die begrenzten Formate (Schlüsselwort 43-49) unterstützen die folgenden Begrenzer:

- / (Schrägstrich)
- - (Strich)
- , (Komma)
- " " (Leerzeichen)
- : (Doppelpunkt)

Monate können als mm oder mmm dargestellt werden; Tage können als d oder dd dargestellt werden; Jahre können als yy oder yyyy dargestellt werden.

**Anmerkung:** Alle Jahre in Datumsangaben können als yyyy statt als yy angegeben werden, um das Jahr 2000 zu unterstützen. Im Interesse der Abwärtskompatibilität wird bei Schlüsselwort 1-16 zu 2-stelligen Jahresangaben yy automatisch 1900 addiert. Bei Schlüsselwort 17-55 wird zu yy < threshold automatisch 2000 addiert; zu yy ≥ threshold wird automatisch 1900 addiert.

**Anmerkung:** Der Jahr-2000-Grenzwert (threshold) wird auf der Registerkarte **Datenbereinigung** des Fensters **Erweiterte Einstellungen** festgelegt (Aufruf über **Optionen > Einstellungen > Erweiterte Einstellungen**).

**Anmerkung:** Wenn Sie den Jahr-2000-Grenzwert ändern, müssen Sie alle Makrofunktionen aktualisieren, die mithilfe der Makrofunktion NUMBER Datumswerte mit 2-stelligen Jahren bearbeiten. Um eine Aktualisierung einer Makrofunktion zu erzwingen, können Sie eine beliebige Bearbeitung vornehmen (z. B. ein Leerzeichen hinzufügen und wieder löschen) und auf das Hakensymbol klicken, damit die Änderung übernommen wird.

**Anmerkung:** Bei Format 0 werden nur die ersten fünf Zeichen jeder Zeichenfolge für die Generierung einer eindeutigen Zahl herangezogen. Alle Zeichenfolgen mit denselben ersten fünf Zeichen ergeben denselben numerischen Wert. Dieselbe Zeichenfolge ergibt jedes Mal denselben numerischen Wert, sogar in verschiedenen Arbeitsblättern. Bearbeiten Sie die Zeichenfolgen bei Bedarf mit Zeichenfolgemakros, sodass die ersten fünf Zeichen eine Klasse eindeutig definieren. Beachten Sie, dass die resultierenden numerischen Werte unter Umständen sehr klein sind. Mithilfe des Fensters **Anzeigeformate** können Sie entweder die Anzahl der angezeigten Dezimalstellen erhöhen oder das Format in den exponentiellen Modus (00E+00) ändern.

## Beispiele

TEMP = NUMBER("\$1.23") oder TEMP = NUMBER("123%", 2)
Erstellt eine neue Spalte TEMP, die den Wert 1.23 enthält.
TEMP = NUMBER(column("Jan", "Mar", "Dec", 8)
Erstellt eine neue Spalte TEMP, die die Werte 1, 3 und 12 enthält.
TEMP = NUMBER("1:52 PM", 14)
Erstellt eine neue Spalte TEMP, die die Zahl 832 enthält.

TEMP = NUMBER("1/1/95", 5)
Erstellt eine neue Spalte TEMP, die die Zahl 728660 enthält.
TEMP = NUMBER(V1)
Erstellt eine neue Spalte TEMP, die die numerischen Werte der Zeichenfolgen in Spalte V1 enthält. Dollarwerte werden korrekt in numerische Werte konvertiert. ??? 's wird bei Zeichenfolgen, die mit dem \$-Format nicht geparkt werden können, zurückgegeben.
TEMP = NUMBER(V1:V3, 4)
Erstellt drei neue Spalten TEMP, VX und VY. Die Spalte TEMP enthält die numerischen Werte der Zeichenfolgen in Spalte V1. Die Spalte VX enthält die numerischen Werte der Zeichenfolgen in Spalte V2. Die Spalte VY enthält die numerischen Werte der Zeichenfolgen in Spalte V3. Alle Datumsangaben in dem Format dd-mmm-yy werden in die Anzahl der Tage seit dem 1. Januar 0000 konvertiert. ??? 's wird bei Zeichenfolgen, die mit dem \$-Format nicht geparkt werden können, zurückgegeben.
TEMP = NUMBER(V1[10:20]:V2, 10)
Erstellt zwei neue Spalten TEMP und VX. Die Spalte TEMP enthält die numerischen Werte der Zeichenfolgen in Zeile 10-20 von Spalte V1. Die Spalte VX enthält die numerischen Werte der Zeichenfolgen in Zeile 10-20 von Spalte V2. Alle standardmäßigen Darstellungen der Wochentage durch drei Zeichen werden in die Zahlen 0 bis 6 konvertiert (0 = Sonntag, 6 = Samstag). Wenn es für einen Wochentag keine Übereinstimmung gibt, wird ??? zurückgegeben.
TEMP = NUMBER(V1, 0)
Sofern Spalte V1 nur 5-stellige Zeichenfolgen enthält, wird eine neue Spalte TEMP erstellt, die für jede eindeutige Zeichenfolge einen unterschiedlichen numerischen Wert enthält.

## Zugehörige Funktionen

Funktion	Beschreibung
WEEKDAY	Konvertiert ASCII-Datumszeichenfolgen in Wochentage

## OR-Makro

Das OR-Makro Makro ist in IBM Campaign und IBM Interact verfügbar.

### Syntax

```
data1 OR data2 data1 || data2
```

### Parameter

data1

Die Zahlen, die durch logisches Oder mit den Werten in data2 verknüpft werden sollen. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

data2

Die Zahl(en), die durch logisches Oder mit den Werten in data1 verknüpft werden soll(en). Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Die Anzahl der Spalten in data2 muss mit der Anzahl der Spalten in data1 übereinstimmen, es sei denn, bei data2 handelt es sich um eine Konstante. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

## Beschreibung

OR berechnet das logische Oder zwischen den beiden angegebenen Datenbereichen. OR gibt für jede Eingabespalte eine neue Spalte zurück, wobei jeweils die entsprechende Spalte von data1 durch logisches Oder mit der entsprechenden Spalte von data2 verknüpft wird (d. h., die erste Spalte von data1 wird durch logisches Oder mit der ersten Spalte von data2 verknüpft, die zweite Spalte mit der zweiten Spalte usw.).

Wenn es sich bei data2 um eine Konstante handelt, wird jeder Wert in data1 durch logisches Oder mit dem betreffenden Wert verknüpft. Wenn data2 eine oder mehrere Spalten enthält, werden die Berechnungen zeilenweise zwischen einer Spalte von data1 und einer Spalte von data2 ausgeführt. Die erste Zeile von data1 wird durch logisches Oder mit der ersten Zeile von data2 verknüpft, die zweite Zeile mit der zweiten Zeile usw. Diese zeilenweise ausgeführte Berechnung erzeugt für jede Zeile ein Ergebnis, bis zum letzten Wert der kürzesten Spalte.

**Anmerkung:** Wenn eine Spalte in jeder Zeile dieselbe Zahl  $x$  wie data2 enthält, ist dies dasselbe, als wenn als data2 die Konstante  $x$  verwendet wird.

**Anmerkung:** Der Operator OR kann durch einen doppelten vertikalen Strich (||) abgekürzt werden. Mit dem doppelten vertikalen Strich können die beiden Argumente getrennt werden (statt z. B. V1 OR 3 anzugeben, können Sie einfach V1||3 eingeben).

## Beispiele

TEMP = 1 OR 8 oder TEMP = 1    8
Erstellt eine neue Spalte TEMP, die den Wert eins enthält (jede Zahl ungleich null wird als eins behandelt).
TEMP = V1    1
Erstellt eine neue Spalte TEMP, die nur Einsen enthält (jeder Wert, der durch logisches Oder mit der Zahl eins verknüpft wird, ergibt eins).
TEMP = V1    V2
Erstellt eine neue Spalte TEMP, in der jeder Zeilenwert von Spalte V1 durch logisches Oder mit dem entsprechenden Zeilenwert von Spalte V2 verknüpft ist.
TEMP = V1:V3    V4:V6
Erstellt drei neue Spalten TEMP, VX und VY. In der Spalte TEMP sind die Werte von Spalte V1 durch logisches Oder mit den entsprechenden Zeilenwerten von Spalte V4 verknüpft. In der Spalte VX sind die Werte von Spalte V2 durch logisches Oder mit den Werten von Spalte V5 verknüpft. In der Spalte VY sind die Werte von Spalte V3 durch logisches Oder mit den Werten von Spalte V6 verknüpft.

```
TEMP = V1[10:20] || V2
```

Erstellt eine neue Spalte TEMP, in deren ersten 11 Zellen die Werte in Zeile 10-20 der Spalten V1 und V2 durch logisches Oder verknüpft sind. Die anderen Zellen in TEMP sind leer.

## Zugehörige Funktionen

Funktion	Beschreibung
UND	Berechnet das logische Und zwischen den beiden angegebenen Datenbereichen.
NOT	Berechnet das logische Nicht des Inhalts des angegebenen Datenbereichs.

---

## POSITION-Makro

Das POSITION-Makro Makro ist nur in IBM Campaign verfügbar.

### Syntax

```
POSITION(colName, pattern [, start [, occurrence]])
```

### Parameter

colName

Der Wert einer Spalte (muss dem Datentyp string angehören).

pattern

Das Muster (die Zeichenfolge), nach dem gesucht wird.

start

Das Byte, bei dem die Suche beginnen soll.

occurrence

Wenn Sie einen Wert für n angeben, wird nach dem n-ten Vorkommen des Musters gesucht.

### Beschreibung

POSITION gibt die Startbyteposition eines Musters oder einer Zeichenfolge innerhalb des Werts einer Spalte (colName), die dem Zeichenfolgedatentyp angehören muss, zurück. Wenn start angegeben ist, beginnt die Suche dort. occurrence ist das n-te Vorkommen des Musters, das zurückgegeben werden soll.

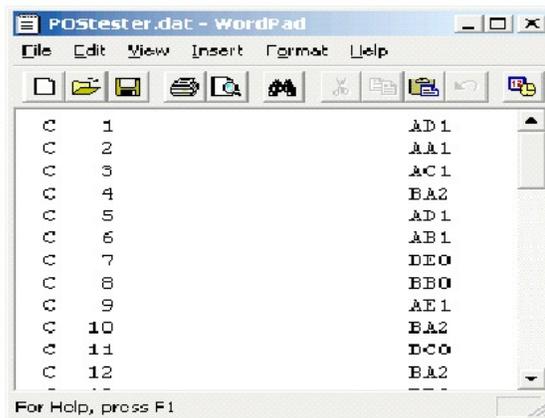
**Anmerkung:** Bei der Suche wird die Groß-/Kleinschreibung nicht beachtet.

### Beispiele

Im folgenden Beispiel wird nach dem Muster oder der Zeichenfolge 'A' innerhalb des Werts der Spalte dbo\_BaseInfo.BranchCd gesucht, der Rückgabewert wird dem abgeleiteten Feld POSstest zugeordnet.



Das folgende Beispiel zeigt einige Zeilen aus der Tabelle, wobei die Werte von `dbo_BaseInfo.BranchCd` und `POStest` nebeneinander zu sehen sind.



Ein komplexeres Beispiel:

```
STRING_SEG(POSITION(CellCode, "X", 1, 2) + 1,
```

```
STRING_LENGTH(Zellencode), Zellencode) = "AAA"
```

Hierdurch werden Zeilen zurückgegeben, in denen die Werte von `CellCode` nach dem zweiten Vorkommen von 'X' am Ende 'AAA' enthalten.

---

## PLUS-Makro

Das PLUS-Makro Makro ist in IBM Campaign und IBM Interact verfügbar.

### Syntax

```
data PLUS addend data + addend
```

### Parameter

```
data
```

Der Zellenbereich mit Zahlen, zu denen etwas addiert werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

addend

Die Zahl(en), die zu allen Werten in der angegebenen Spalte addiert werden soll(en). Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Die Anzahl der Spalten in data2 muss mit der Anzahl der Spalten in data1 übereinstimmen, es sei denn, bei data2 handelt es sich um eine Konstante. Informationen zur Formatdefinition von addend (identisch mit data) finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

## Beschreibung

PLUS addiert die Werte in den beiden angegebenen Datenbereichen. PLUS gibt für jede Eingabespalte eine neue Spalte zurück, wobei jeweils zu der entsprechenden Spalte von data1 die entsprechende Spalte von data2 addiert wird (d. h., zur ersten Spalte von data wird die erste Spalte von data1 addiert, zur zweiten Spalte die zweite Spalte usw.).

Wenn es sich bei data2 um eine Konstante handelt, wird jeder Wert in data1 um den betreffenden Wert erhöht. Wenn data2 eine oder mehrere Spalten enthält, werden die Berechnungen zeilenweise zwischen einer Spalte von data2 und einer Spalte von data2 ausgeführt. Zur ersten Zeile von data1 wird die erste Zeile von data2 addiert, zur zweiten Zeile die zweite Zeile usw. Diese zeilenweise ausgeführte Berechnung erzeugt für jede Zeile ein Ergebnis, bis zum letzten Wert der kürzesten Spalte.

**Anmerkung:** Der Operator PLUS kann durch ein Pluszeichen (+) abgekürzt werden.

## Beispiele

TEMP = 3 PLUS 4 oder TEMP = 3 + 4
Erstellt eine neue Spalte TEMP, die den Wert sieben enthält.
TEMP = V1 + 8
Erstellt eine neue Spalte TEMP, in der jeder Wert den Inhalt von Spalte V1 plus acht darstellt.
TEMP = V1 + V1
Erstellt eine neue Spalte TEMP, die jeweils den zweifachen Inhalt von Spalte V1 enthält.
TEMP = V1 + V2
Erstellt eine neue Spalte TEMP, wobei jeder Wert den Zeilenwert von Spalte V1 plus den entsprechenden Zeilenwert von Spalte V2 darstellt.
TEMP = V1:V3 + V4:V6
Erstellt drei neue Spalten TEMP, VX und VY. Die Spalte TEMP enthält die Werte von Spalte V1 plus die entsprechenden Zeilenwerte von Spalte V4. Die Spalte VX addiert Spalte V2 und Spalte V5. Die Spalte VY addiert Spalte V3 und Spalte V6.

TEMP = V1[10:20] + V2 oder TEMP = V1[10:20] + V2[1:11]

Erstellt eine neue Spalte TEMP, deren erste 11 Zellen das Ergebnis der Addition der Werte in Zeile 10-20 von Spalte V1 und der Werte in Zeile 1-11 von Spalte V2 enthalten. Die anderen Zellen in TEMP sind leer.

## Zugehörige Funktionen

Funktion	Beschreibung
MINUS	Subtrahiert einen angegebenen Datenbereich von einem anderen.
SUM oder TOTAL	Berechnet die Summe eines Zellenbereichs.

---

## POW-Makro

Das POW-Makro Makro ist in IBM Campaign und IBM Interact verfügbar.

### Syntax

base POW exponent base ^ exponent

### Parameter

base

Die numerischen Werte, die in die Potenz eines Exponenten erhoben werden sollen. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von base (identisch mit data) finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

exponent

Die exponentielle(n) Zahl(en), in deren Potenz die Werte in data erhoben werden sollen. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Die Anzahl der Spalten in exponent muss mit der Anzahl der Spalten in base übereinstimmen, es sei denn, bei base handelt es sich um eine Konstante. Informationen zur Formatdefinition von exponent (identisch mit data) finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

### Beschreibung

POW erhebt die Werte im ersten Datenbereich in die im zweiten Datenbereich angegebene Potenz (berechnet wird also  $base^{exponent}$ ). POW gibt für jede Eingabespalte eine neue Spalte zurück, die jeweils das Ergebnis der Erhebung von base in die Potenz von exponent enthält (d. h., die erste Spalte von data1 wird in die Potenz der ersten Spalte von data erhoben, die zweite Spalte in die der zweiten Spalte usw.).

Wenn es sich bei exponent um eine Konstante handelt, wird jeder Wert in base in die betreffende Potenz erhoben. Wenn exponent eine oder mehrere Spalten enthält, werden die Berechnungen zeilenweise zwischen einer Spalte von base und einer

Spalte von exponent ausgeführt. Die erste Zeile von base wird in die Potenz der ersten Zeile von exponent erhoben, die zweite Zeile in die der zweiten Zeile usw. Diese zeilenweise ausgeführte Berechnung erzeugt für jede Zeile ein Ergebnis, bis zum letzten Wert der kürzesten Spalte.

**Anmerkung:** Der Operator POW kann durch einen Zirkumflex (^) abgekürzt werden. Beispielsweise ist  $TEMP = 2^8$  äquivalent zu  $TEMP = 2 \text{ POW } 8$ .

**Anmerkung:** Wenn der Wert  $x$  zu groß oder zu klein ist, wird ein Überlauffehler zurückgegeben. Dies ist der Fall, wenn  $base^{\text{exponent}}$  den maximalen oder minimalen 32-Bit-Gleitkommawert überschreitet.

## Beispiele

$TEMP = 2 \text{ POW } 3$ oder $TEMP = 2^3$ Erstellt eine neue Spalte TEMP, die den Wert acht enthält.
$TEMP = V1 \wedge 0.5$ Erstellt eine neue Spalte TEMP, in der jeder Wert die Quadratwurzel des Inhalts von Spalte V1 darstellt (äquivalent zu $SQRT(V1)$ ).
$TEMP = V1 \wedge V3$ Erstellt eine neue Spalte TEMP, wobei jeder Zeilenwert von Spalte V1 in die Potenz des entsprechenden Zeilenwerts von Spalte V2 erhoben wird.
$TEMP = V1:V3 \wedge V4:V6$ Erstellt drei neue Spalten TEMP, VX und VY. In der Spalte TEMP sind die Werte von Spalte V1 in die Potenz der entsprechenden Zeilenwerte von Spalte V4 erhoben. Die Spalte VX enthält das Ergebnis der Erhebung von Spalte V2 in die durch die entsprechenden Werte von Spalte V5 angegebene Potenz. Die Spalte VY enthält das Ergebnis der Erhebung von Spalte V3 in die durch die entsprechenden Werte von Spalte V6 angegebene Potenz.
$TEMP = V1[10:20] \text{ POW } V2$ oder $TEMP = V1[10:20] \text{ POW } V2[1:11]$ Erstellt eine neue Spalte TEMP, deren erste 11 Zellen das Ergebnis der Erhebung der Werte in Zeile 10-20 von Spalte V1 in die durch die Werte in Zeile 1-10 von Spalte V2 angegebene Potenz enthalten. Die anderen Zellen in TEMP sind leer.

## Zugehörige Funktionen

Funktion	Beschreibung
EXP	Erhebt die natürliche Zahl (e) in die Potenz, die durch den Inhalt jeder Zelle im angegebenen Datenbereich angegeben ist.
LN oder LOG	Berechnet den natürlichen Logarithmus des Inhalts des angegebenen Datenbereichs.
LN2	Berechnet den binären Logarithmus des Inhalts des angegebenen Datenbereichs.
LN10	Berechnet den dekadischen Logarithmus des Inhalts des angegebenen Datenbereichs.

## RANDOM-Makro

Das RANDOM-Makro Makro ist nur in IBM Campaign verfügbar.

## Syntax

RANDOM(num [, seed]) RANDOM(num, value1, value2 [, seed])

## Parameter

num

Die Anzahl der Zufallszahlen, die generiert werden sollen. Dieser Wert muss eine positive Ganzzahl größer als null sein.

value1

Eine Grenze für die Zufallszahlen, die generiert werden sollen. Dabei kann es sich um einen konstanten Wert oder um einen Ausdruck handeln, dessen Auswertung eine Konstante ergibt. Wird dieser Parameter nicht angegeben, ist der Standard null.

value2

Die andere Grenze für die Zufallszahlen, die generiert werden sollen. Dabei kann es sich um einen konstanten Wert oder um einen Ausdruck handeln, dessen Auswertung eine Konstante ergibt. Wird dieser Parameter nicht angegeben, ist der Standard eins.

seed

Ein optionaler Seed für die Generierung von Zufallszahlen. Der Wert muss eine ganze Zahl sein.

## Beschreibung

RANDOM generiert eine Spalte von Zufallszahlen. Dabei wird eine neue Spalte mit num Zufallszahlen zurückgegeben. Wenn value1 und value2 angegeben sind, werden die Zufallszahlen zwischen diesen Grenzen (inklusive) generiert. Wenn sie nicht angegeben sind, werden standardmäßig Werte zwischen null und eins generiert. Wenn seed angegeben ist, wird dieser Wert als Seed für die Generierung von Zufallszahlen verwendet.

**Anmerkung:** Wenn seed größer-gleich  $2^{32}$  ist, wird der Wert durch  $2^{32}-1$  ersetzt. Werte von seed über  $2^{24}$  werden gerundet (d. h., die Genauigkeit geht verloren). Deshalb können sich bei demselben Wert von seed mehrere Werte ergeben.

## Beispiele

TEMP = RANDOM() Erstellt eine neue Spalte TEMP, die Zufallszahlen mit unbegrenzter Länge enthält.
TEMP = RANDOM(100) Erstellt eine neue Spalte TEMP, die 100 Zufallszahlen zwischen 0,0 und 1,0 enthält.
TEMP = RANDOM(100, 5943049) Erstellt eine neue Spalte TEMP mit 100 Zufallszahlen, die ausgehend von dem Seedwert 5943049 generiert werden.

TEMP = RANDOM(100, 0, 100)
Erstellt eine neue Spalte TEMP, die 100 Zufallszahlen zwischen 0 und 100,0 enthält.
TEMP = RANDOM(100, 0, 100, 5943049)
Erstellt eine neue Spalte TEMP mit 100 Zufallszahlen zwischen 0 und 100, die ausgehend von dem Seedwert 5943049 generiert werden.

## Zugehörige Funktionen

Funktion	Beschreibung
RANDOM_GAUSS	Gibt die angegebene Anzahl von Zufallswerten aus einer gaußschen Verteilung zurück.

---

## RANDOM\_GAUSS-Makro

Das RANDOM\_GAUSS-Makro Makro ist nur in IBM Campaign verfügbar.

### Syntax

RANDOM\_GAUSS(num [, seed]) RANDOM\_GAUSS(num, mean, std [, seed])

### Parameter

num

Die Anzahl der Zufallszahlen, die generiert werden sollen. Dieser Wert muss eine positive Ganzzahl größer als null sein.

mean

Der Mittelwert der gaußschen Kurve. Dabei kann es sich um einen konstanten Wert oder um einen Ausdruck handeln, dessen Auswertung eine Konstante ergibt. Wird dieser Parameter nicht angegeben, ist der Standard null.

std

Die Standardabweichung der gaußschen Kurve. Dabei kann es sich um einen konstanten Wert oder um einen Ausdruck handeln, dessen Auswertung eine Konstante ergibt. Wird dieser Parameter nicht angegeben, ist der Standard eins.

seed

Ein optionaler Seed für die Generierung von Zufallszahlen. Der Wert muss eine ganze Zahl sein. (Wenn der angegebene Wert keine Ganzzahl ist, wird er automatisch durch die Untergrenze des Werts ersetzt.)

### Beschreibung

RANDOM\_GAUSS generiert eine Spalte von Zufallszahlen, der eine gaußsche Verteilung zugrunde liegt. Dabei wird eine neue Spalte mit num Zufallszahlen zurückgegeben. Wenn mean und std angegeben sind, wird bei der Generierung der Zufallszahlen eine gaußsche Verteilung mit diesem Mittelwert und dieser Standardabweichung

zugrunde gelegt. Wenn sie nicht angegeben sind, hat die gaußsche Verteilung standardmäßig den Mittelwert null und die Standardabweichung eins. Wenn seed angegeben ist, wird dieser Wert als Seed für die Generierung von Zufallszahlen verwendet.

## Beispiele

TEMP = RANDOM_GAUSS(100)
Erstellt eine neue Spalte TEMP mit 100 Werten, die zufällig aus einer gaußschen Verteilung mit Mittelwert null und Einheitsstandardabweichung gewonnen wurden.
TEMP = RANDOM_GAUSS(500, 3)
Erstellt eine neue Spalte TEMP mit 100 Werten, die zufällig aus einer gaußschen Verteilung mit Mittelwert null und Einheitsstandardabweichung gewonnen wurden. Die Zahl 3 wird als Generierungswert für den Zufallsgenerator verwendet.
TEMP = RANDOM_GAUSS(5000, 100, 32)
Erstellt eine neue Spalte TEMP mit 5000 Werten, die zufällig aus einer gaußschen Verteilung mit Mittelwert 100 und Standardabweichung 32 gewonnen wurden.
TEMP = RANDOM_GAUSS(500, -1, 2, 3)
Erstellt eine neue Spalte TEMP mit 500 Werten, die zufällig aus einer gaußschen Verteilung mit Mittelwert -1 und Standardabweichung 2 gewonnen wurden. Die Zahl 3 wird als Generierungswert für den Zufallsgenerator verwendet.

## Zugehörige Funktionen

Funktion	Beschreibung
RANDOM	Gibt die angegebene Anzahl von Zufallszahlen zurück.

---

## ROUND-Makro

Das ROUND-Makro Makro ist nur in IBM Campaign verfügbar.

### Syntax

ROUND(data)

### Parameter

data

Die numerischen Werte, die gerundet werden sollen. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

### Beschreibung

ROUND rundet die Werte im angegebenen Datenbereich auf die nächste Ganzzahl. Dabei wird für jede Eingabespalte eine neue Spalte zurückgegeben, die den gerundeten Wert der Zahlen in der entsprechenden Eingabespalte enthält. Genau in der

Mitte liegende Zahlen werden aufgerundet (so wird 2.5 zu 3.0 und -2.5 zu -2.0 gerundet).

## Beispiele

TEMP = ROUND(3.2)
Erstellt eine neue Spalte TEMP, die den Wert drei enthält.
TEMP = ROUND(V1)
Erstellt eine neue Spalte TEMP, in der jeder Wert den gerundeten Wert des Inhalts von Spalte V1 darstellt.
TEMP = ROUND(V1:V3)
Erstellt drei neue Spalten TEMP, VX und VY. Die Werte in der Spalte TEMP sind die gerundeten Werte des Inhalts von Spalte V1, die Werte von Spalte VX sind die gerundeten Werte des Inhalts von Spalte V2 und die Werte von Spalte VY sind die gerundeten Werte des Inhalts von Spalte V3.
TEMP = ROUND(V1[10:20])
Erstellt eine neue Spalte TEMP, in der die ersten 11 Zellen die gerundeten Werte in Zeile 10-20 von Spalte V1 enthalten. Die anderen Zellen in TEMP sind leer.
TEMP = ROUND(V1[1:5]:V2)
Erstellt zwei neue Spalten TEMP und VX, die jeweils die Werte in Zeile 1-5 enthalten (die anderen Zellen sind leer). Die Werte in der Spalte TEMP sind die gerundeten Werte der entsprechenden Zeilen von Spalte V1, die Werte in der Spalte VX sind die gerundeten Werte der entsprechenden Zeilen von Spalte V2.

## Zugehörige Funktionen

Funktion	Beschreibung
INT	Berechnet den abgerundeten Ganzzahlwert des Inhalts des angegebenen Datenbereichs.
MOD	Berechnet den Modulo-Wert des Inhalts des angegebenen Datenbereichs.
TRUNCATE	Berechnet den Ganzzahlanteil jedes Werts im angegebenen Datenbereich.

## ROWNUM-Makro

Das ROWNUM-Makro Makro ist nur in IBM Campaign verfügbar.

### Syntax

ROWNUM()

### Beschreibung

ROWNUM generiert fortlaufende Zahlen von eins bis zur Anzahl der Einträge. Die Zahl für den ersten Datensatz ist eins, die Zahl für den zweiten Datensatz ist zwei usw.

**Anmerkung:** ROWNUM kann maximal zwei Milliarden Einträge verarbeiten.

---

## RTRIM-Makro

Das RTRIM-Makro ist in IBM Campaign und IBM Interact verfügbar.

### Syntax

RTRIM(data)

### Parameter

data

### Beschreibung

RTRIM entfernt nachfolgende Leerzeichen aus jedem Zeichenfolgewert im angegebenen Datenbereich und gibt die konvertierte Zeichenfolge zurück. RTRIM gibt für jede Eingabespalte eine neue Spalte zurück.

### Beispiele

```
Temp = RTRIM "gold "
```

Erstellt eine neue Zeichenfolge Temp, die "gold" enthält.

---

## SIGN-Makro

Das SIGN-Makro Makro ist nur in IBM Campaign verfügbar.

### Syntax

SIGN(data)

Parameter

data

Die numerischen Werte, deren Vorzeichen berechnet werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

### Beschreibung

SIGN testet das Vorzeichen der Werte im angegebenen Datenbereich. SIGN gibt für jede Eingabespalte eine neue Spalte zurück, die das Vorzeichen der Zahlen in der entsprechenden Eingabespalte enthält. Bei allen Werten größer als null wird eine positive Eins zurückgegeben; bei allen Werten kleiner als null wird eine negative Eins zurückgegeben; bei allen Werten gleich null wird eine Null zurückgegeben.

### Beispiele

```
TEMP = SIGN(-3)
```

Erstellt eine neue Spalte TEMP, die den Wert -1 enthält.

TEMP = SIGN(V1)
Erstellt eine neue Spalte TEMP, in der jeder Wert das Vorzeichen des Inhalts von Spalte V1 darstellt.
TEMP = SIGN(V1:V3)
Erstellt drei neue Spalten TEMP, VX und VY. Die Werte in der Spalte TEMP sind die Vorzeichen des Inhalts von Spalte V1, die Werte von Spalte VX sind die Vorzeichen des Inhalts von Spalte V2 und die Werte von Spalte VY sind die Vorzeichen des Inhalts von Spalte V3.
TEMP = SIGN(V1[10:20])
Erstellt eine neue Spalte TEMP, in der die ersten 11 Zellen jeweils die Vorzeichen der Werte in Zeile 10-20 von Spalte V1 enthalten. Die anderen Zellen in TEMP sind leer.
TEMP = SIGN(V1[10:50]:V2)
Erstellt zwei neue Spalten TEMP und VX, die jeweils die Werte in Zeile 1-41 enthalten (die anderen Zellen sind leer). Die Werte in der Spalte TEMP sind die Vorzeichen der Werte in Zeile 10-50 von Spalte V1, die Werte in der Spalte VX sind die Vorzeichen der Werte in Zeile 10-50 von Spalte V2.

## Zugehörige Funktionen

Funktion	Beschreibung
ABS	Berechnet den absoluten Wert des Inhalts des angegebenen Datenbereichs.
INVERSE	Berechnet den Negativwert des Inhalts des angegebenen Datenbereichs.

---

## SIN-Makro

Das SIN-Makro Makro ist nur in IBM Campaign verfügbar.

### Syntax

SIN(data [, units\_keyword])

### Parameter

data

Die numerischen Werte, deren Sinus berechnet werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

units\_keyword

Dieses optionale Schlüsselwort legt fest, ob die Eingabewerte und Ergebnisse als Grad oder als Radiant interpretiert werden. Wählen Sie eine der folgenden Angaben aus:

RADIAN - Führt die Berechnungen in Radiant aus (Standard)

DEGREE - Führt die Berechnungen in Grad aus

Wird dieser Parameter nicht angegeben, ist der Standard Radiant. (Die Umrechnung von Radiant in Grad erfolgt durch Division durch PI und Multiplikation mit 180.)

Weitere Informationen zur Verwendung von Schlüsselwörtern in IBM Campaign finden Sie in „Formatspezifikationen“ auf Seite 9.

## Beschreibung

SIN berechnet den Sinus der Werte im angegebenen Datenbereich. SIN gibt für jede Eingabespalte eine neue Spalte zurück, die den Sinus der Zahlen in der entsprechenden Eingabespalte enthält.

## Beispiele

TEMP = SIN(PI/2) oder TEMP = SIN(PI/2, 0) oder TEMP = SIGN(PI/2, RADIAN)
Erstellt eine neue Spalte TEMP, die den Wert eins enthält.
TEMP = SIN(V1)
Erstellt eine neue Spalte TEMP, in der jeder Wert den Sinus (in Radiant) des Inhalts von Spalte V1 darstellt.
TEMP = SIN(V1:V3, 1) oder TEMP = SIN(V1:V3, DEGREE)
Erstellt drei neue Spalten TEMP, VX und VY. Die Werte in der Spalte TEMP sind jeweils der Sinus des Inhalts von Spalte V1, die Werte von Spalte VX sind der Sinus des Inhalts von Spalte V2 und die Werte von Spalte VY sind der Sinus des Inhalts von Spalte V3. Alle Werte sind in Grad angegeben.
TEMP = SIN(V1[10:50]:V2)
Erstellt zwei neue Spalten TEMP und VX, die jeweils die Werte in Zeile 1-41 enthalten (die anderen Zellen sind leer). Die Werte in der Spalte TEMP sind der Sinus der Werte in Zeile 10-50 von Spalte V1, die Werte in der Spalte VX sind der Sinus der Werte in Zeile 10-50 von Spalte V2. Alle Werte sind in Radiant angegeben.

## Zugehörige Funktionen

Funktion	Beschreibung
ASIN	Berechnet den Arkussinus des Inhalts des angegebenen Datenbereichs.
COS	Berechnet den Kosinus des Inhalts des angegebenen Datenbereichs.
SINH	Berechnet den Hyperbelsinus des Inhalts des angegebenen Datenbereichs.
TAN	Berechnet den Tangens des Inhalts des angegebenen Datenbereichs.

---

## SINH-Makro

Das SINH-Makro Makro ist nur in IBM Campaign verfügbar.

## Syntax

SINH(data [, units\_keyword])

## Parameter

data

Die numerischen Werte, deren Hyperbelsinus berechnet werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

units\_keyword

Dieses optionale Schlüsselwort legt fest, ob die Eingabewerte und Ergebnisse als Grad oder als Radiant interpretiert werden. Wählen Sie eine der folgenden Angaben aus:

RADIAN - Führt die Berechnungen in Radiant aus (Standard)

DEGREE - Führt die Berechnungen in Grad aus

Wird dieser Parameter nicht angegeben, ist der Standard Radiant. (Die Umrechnung von Radiant in Grad erfolgt durch Division durch PI und Multiplikation mit 180.)

Weitere Informationen zur Verwendung von Schlüsselwörtern in IBM Campaign finden Sie in „Formatspezifikationen“ auf Seite 9.

## Beschreibung

SINH berechnet den Hyperbelsinus der Werte im angegebenen Datenbereich. SINH gibt für jede Eingabespalte eine neue Spalte zurück, die den Hyperbelsinus der Zahlen in der entsprechenden Eingabespalte enthält. Bei  $x$  in Radiant ist der Hyperbelsinus einer Zahl

$$\sinh(x) = \frac{e^x - e^{-x}}{2}$$

Dabei ist  $e$  die natürliche Zahl, 2,7182818.

**Anmerkung:** Wenn der Wert  $x$  zu groß ist, wird ein Überlauffehler zurückgegeben. Dies geschieht, wenn  $\sinh(x)$  den maximalen 32-Bit-Gleitkommawert überschreitet.

## Beispiele

```
TEMP = SINH(1) oder TEMP = SINH(1, 0) oder TEMP = SINH(1, RADIAN)
```

Erstellt eine neue Spalte TEMP, die den Wert 1.18 enthält.

TEMP = SINH(V1)
Erstellt eine neue Spalte TEMP, in der jeder Wert den Hyperbelsinus (in Radiant) des Inhalts von Spalte V1 darstellt.
TEMP = SINH(V1:V3, 1) oder TEMP = SINH(V1:V3, DEGREE)
Erstellt drei neue Spalten TEMP, VX und VY. Die Werte in der Spalte TEMP sind jeweils der Hyperbelsinus des Inhalts von Spalte V1, die Werte von Spalte VX sind der Hyperbelsinus des Inhalts von Spalte V2 und die Werte von Spalte VY sind der Hyperbelsinus des Inhalts von Spalte V3. Alle Werte sind in Grad angegeben.
TEMP = SINH(V1[10:50]:V2)
Erstellt zwei neue Spalten TEMP und VX, die jeweils die Werte in Zeile 1-41 enthalten (die anderen Zellen sind leer). Die Werte in der Spalte TEMP sind der Hyperbelsinus der Werte in Zeile 10-50 von Spalte V1, die Werte in der Spalte VX sind der Hyperbelsinus der Werte in Zeile 10-50 von Spalte V2. Alle Werte sind in Radiant angegeben.

## Zugehörige Funktionen

Funktion	Beschreibung
COSH	Berechnet den Hyperbelkosinus des Inhalts des angegebenen Datenbereichs.
SIN	Berechnet den Sinus des Inhalts des angegebenen Datenbereichs.
TANH	Berechnet den Hyperbeltangens des Inhalts des angegebenen Datenbereichs.

## SQRT-Makro

Das SQRT-Makro Makro ist nur in IBM Campaign verfügbar.

### Syntax

SQRT(data)

Parameter

data

Die numerischen Werte, deren Quadratwurzel berechnet werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

### Beschreibung

SQRT berechnet die Quadratwurzel der Werte im angegebenen Datenbereich. SQRT gibt für jede Eingabespalte eine neue Spalte zurück, die die positive Quadratwurzel der Zahlen in der entsprechenden Eingabespalte enthält.

**Anmerkung:** Wenn ein Wert im definierten Datenbereich negativ ist, wird der Wert ??? für diese Zelle zurückgegeben.

## Beispiele

TEMP = SQRT(2)
Erstellt eine neue Spalte TEMP, die den Wert 1.41 enthält.
TEMP = SQRT(V1)
Erstellt eine neue Spalte TEMP, in der jeder Wert die Quadratwurzel des Inhalts von Spalte V1 darstellt.
TEMP = SQRT(V1:V3)
Erstellt drei neue Spalten TEMP, VX und VY. Die Werte in der Spalte TEMP sind jeweils die Quadratwurzeln des Inhalts von Spalte V1, die Werte von Spalte VX sind die Quadratwurzeln des Inhalts von Spalte V2 und die Werte von Spalte VY sind die Quadratwurzeln des Inhalts von Spalte V3.
TEMP = SQRT(V1[10:20])
Erstellt eine neue Spalte TEMP, in der die ersten 11 Zellen jeweils die Quadratwurzeln der Werte in Zeile 10-20 von Spalte V1 enthalten. Die anderen Zellen in TEMP sind leer.
TEMP = SQRT(V1[10:50]:V2)
Erstellt zwei neue Spalten TEMP und VX, die jeweils die Werte in Zeile 1-41 enthalten (die anderen Zellen sind leer). Die Werte in der Spalte TEMP sind die Quadratwurzeln der Werte in Zeile 10-50 von Spalte V1, die Werte in der Spalte VX sind die Quadratwurzeln der Werte in Zeile 10-50 von Spalte V2.

## Zugehörige Funktionen

Funktion	Beschreibung
DIV	Dividiert einen angegebenen Datenbereich durch einen anderen.
MULT	Multipliziert die Inhalte zweier Datenbereiche.
POW	Erhebt einen Basiswert in die Potenz des oder der angegebenen Exponenten.

---

## STDV- oder STDEV-Makro

Das STDV- oder STDEV-Makro Makro ist in IBM Campaign und IBM Interact verfügbar.

### Syntax

STDV(data [, keyword]) STDEV(data [, keyword])

### Parameter

data

Die numerischen Werte, deren Standardabweichung berechnet werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

keyword

Dieses optionale Schlüsselwort legt fest, wie die Berechnung auf den Eingabebereich angewendet wird. Wählen Sie eine der folgenden Angaben aus:

ALL - Wendet die Berechnung auf alle Zellen in data an (Standard)

COL - Führt die Berechnung für jede Spalte von data gesondert aus

ROW - Führt die Berechnung für jede Zeile von data gesondert aus

Weitere Informationen zur Verwendung von Schlüsselwörtern in IBM Campaign finden Sie in „Formatspezifikationen“ auf Seite 9.

**Anmerkung:** Viele Makrofunktionen setzen die Schlüsselwortparameter {ALL | COL | ROW} ein. Diese Schlüsselwörter gelten nicht für **IBM Campaign**, da es sich bei den Eingabedaten immer um eine Einzelspalte oder ein Einzelfeld handelt. Das Makro verhält sich immer so, als ob das Schlüsselwort COL angegeben würde. Daher brauchen Sie diese Schlüsselwörter nicht anzugeben, wenn Sie **IBM Campaign** verwenden.

## Beschreibung

STDV berechnet die Standardabweichung aller Zellen im angegebenen Datenbereich. Die Standardabweichung einer Verteilung ist die Quadratwurzel der Varianz. Die Standardabweichung wird wie folgt berechnet:

$$\sqrt{\frac{1}{n-1} \sum_{j=1}^n (x_j - \text{mean})^2}$$

Dabei steht  $x$  für eine Stichprobe,  $n$  für die Anzahl der Stichproben und *Mittelwert* für den Durchschnitt der Verteilung.

**Anmerkung:** Wenn die Anzahl der Stichproben  $n = 1$  ist, gibt STDV einen Fehler zurück.

## Beispiele

TEMP = STDV(V1)

Erstellt eine neue Spalte TEMP, die einen Einzelwert enthält, der die Standardabweichung des Inhalts von Spalte V1 darstellt.

TEMP = STDV(V1:V3)

Erstellt eine neue Spalte TEMP, die einen Einzelwert enthält, der die Standardabweichung des Inhalts der Spalten V1, V2 und V3 darstellt.

TEMP = STDV(V1[1:5]:V4)

Erstellt eine neue Spalte TEMP, die einen Einzelwert enthält, der die Standardabweichung der Zellen in Zeile 1-5 von Spalte V1 bis V4 darstellt.

TEMP = STDV(V1:V3, COL)
Erstellt drei neue Spalten TEMP, VX und VY. Der Einzelwert in der Spalte TEMP ist die Standardabweichung des Inhalts von Spalte V1, der Einzelwert in der Spalte VX ist die Standardabweichung des Inhalts von Spalte V2 und der Einzelwert in der Spalte VY ist die Standardabweichung des Inhalts von Spalte V3.
TEMP = STDV(V1[10:50]:V3, COL)
Erstellt drei neue Spalten TEMP, VX und VY, die jeweils einen Einzelwert enthalten. Der Wert in der Spalte TEMP ist die Standardabweichung der Zellen in Zeile 10-50 von Spalte V1, der Wert in der Spalte VX ist die Standardabweichung der Zellen in Zeile 10-50 von Spalte V2 und der Wert in der Spalte VY ist die Standardabweichung der Zellen in Zeile 10-50 von Spalte V3.
TEMP = STDV(V1:V3, ROW)
Erstellt eine neue Spalte TEMP, in der jeder Zelleneintrag die Standardabweichung der entsprechenden Zeile in Spalte V1, V2 und V3 darstellt.
TEMP = STDV(V1[1:5]:V3, ROW)
Erstellt eine neue Spalte TEMP, in der die Zellen in Zeile 1-5 die Standardabweichung der entsprechenden Zeile in Spalte V1 bis V3 enthalten. Die anderen Zellen in TEMP sind leer.

## Zugehörige Funktionen

Funktion	Beschreibung
VAR	Berechnet die Varianz eines Zellenbereichs

---

## STRING\_CONCAT-Makro

Das STRING\_CONCAT-Makro Makro ist in IBM Campaign und IBM Interact verfügbar.

### Syntax

STRING\_CONCAT(string1, string2, ... stringN)

### Parameter

string

Eine ASCII-Zeichenfolge, die verkettet werden soll. Dabei kann es sich um ASCII-Text in Anführungszeichen, eine Textspalte, einen Zellenbereich mit Text oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Der Abschnitt *Makrofunktionsparameter* im Kapitel *Verwendung von Makros* für Ihr Produkt enthält Informationen zur Formatdefinition von string (identisch mit der von data).

### Beschreibung

STRING\_CONCAT verkettet die ASCII-Textwerte in den angegebenen Datenbereichen. Dabei wird für jede Eingabespalte eine neue Spalte zurückgegeben, die die verketteten Zeichenfolgen aus den entsprechenden Zeilen von string enthält. Diese zeilenweise ausgeführte Berechnung erzeugt für jede Zeile ein Ergebnis, bis zum letzten Wert der kürzesten Spalte.

**Anmerkung:** Die Gesamtbreite jeder Ergebniszeichenfolge darf 255 Zeichen nicht überschreiten.

IBM Interact unterstützt auch die folgende Syntax:

`STRING_CONCAT( string1 , string2 , ... stringN )`

Zum Beispiel ist `STRING_CONCAT('a', 'b', 'c', 'd')` gültig.

## Beispiele

<code>TEMP = STRING_CONCAT("house", "boat")</code>
Erstellt eine neue Spalte TEMP, die die ASCII-Zeichenfolge "houseboat" enthält.
<code>TEMP = STRING_CONCAT(V1, ".")</code>
Erstellt eine neue Spalte TEMP, in der jede Zeile die ASCII-Zeichenfolge in der entsprechenden Zeile von Spalte V1 sowie einen angehängten Punkt enthält.
<code>TEMP = STRING_CONCAT(V1, V2)</code>
Erstellt eine neue Spalte TEMP, wobei in jeder Zeile die ASCII-Zeichenfolge in Spalte V1 mit der Zeichenfolge in Spalte V2 verkettet ist.
<code>TEMP = STRING_CONCAT(V1:V3, V4:V6)</code>
Erstellt drei neue Spalten TEMP, VX und VY. Die Werte in der Spalte TEMP sind die verketteten Zeichenfolgen aus den entsprechenden Zeilen der Spalten V1 und V4, die Werte der Spalte VX sind die verketteten Zeichenfolgen aus den entsprechenden Zeilen der Spalten V2 und V5 und die Werte der Spalte VY sind die verketteten Zeichenfolgen aus den entsprechenden Zeilen der Spalten V3 und V6.
<code>TEMP = STRING_CONCAT(V1[5:10]:V2, V3:V4)</code>
Erstellt zwei neue Spalten TEMP und VX. Die Werte in der Spalte TEMP sind Zeichenfolgen aus den Zeilen 5-10 von Spalte V1, die mit den Zeichenfolgen aus den Zeilen 1-6 von Spalte V3 verkettet sind. Die Werte in der Spalte VX sind Zeichenfolgen aus den Zeilen 5-10 von Spalte V2, die mit den Zeichenfolgen aus den Zeilen 1-6 von Spalte V4 verkettet sind.
<code>TEMP = STRING_CONCAT('a', 'b', 'c', 'd')</code>
Erstellt eine neue Spalte TEMP, die die ASCII-Zeichenfolge "abcd" enthält.

## Zugehörige Funktionen

Funktion	Beschreibung
STRING_HEAD	Gibt die ersten n Zeichen jeder Zeichenfolge im angegebenen Datenbereich zurück.
STRING_LENGTH	Gibt die Länge jeder Zeichenfolge im angegebenen Datenbereich zurück
STRING_SEG	Gibt das Zeichenfolgesegment zwischen zwei angegebenen Indizes zurück
STRING_TAIL	Gibt die letzten n Zeichen jeder Zeichenfolge im angegebenen Datenbereich zurück.

## STRING\_HEAD-Makro

Das STRING\_HEAD-Makro Makro ist nur in IBM Campaign verfügbar.

### Syntax

`STRING_HEAD(num_chars, data)`

## Parameter

num\_chars

Die Anzahl der Zeichen, die vom Anfang jeder Zeichenfolge in data zurückgegeben werden sollen. Dieser Wert muss eine positive Ganzzahl größer als null sein.

data

ASCII-Zeichenfolgewerte. Dabei kann es sich um ASCII-Text in Anführungszeichen, eine Textspalte, einen Zellenbereich mit Text oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

## Beschreibung

STRING\_HEAD gibt die ersten num\_chars-Zeichen jedes Zeichenfolgewerts im angegebenen Datenbereich zurück. Wenn num\_chars größer als die Anzahl der Zeichen in einer Zeichenfolge ist, werden die verbleibenden Zeichen mit dem Nullzeichen "\0" aufgefüllt.

## Beispiele

TEMP = STRING_HEAD(3, "JAN 15, 1997")
Erstellt eine neue Spalte TEMP, die die ASCII-Zeichenfolge "JAN" enthält.
TEMP = STRING_HEAD(10, "Pressure")
Erstellt eine neue Spalte TEMP, die die ASCII-Zeichenfolge "Pressure" enthält.
TEMP = STRING_HEAD(5, V1)
Erstellt eine neue Spalte TEMP, die die ersten fünf Zeichen jeder Zeichenfolge in der Spalte V1 enthält.
TEMP = STRING_HEAD(1, V1:V3)
Erstellt drei neue Spalten TEMP, VX und VY. Die Werte in der Spalte TEMP sind jeweils das erste Zeichen der Zeichenfolgen in den entsprechenden Zeilen von Spalte V1, die Werte von Spalte VX sind das erste Zeichen der Zeichenfolgen in den entsprechenden Zeilen von Spalte V2 und die Werte von Spalte VY sind das erste Zeichen der Zeichenfolgen in den entsprechenden Zeilen von Spalte V3.
TEMP = STRING_HEAD(12, V4[1:50]:V6]
Erstellt drei neue Spalten TEMP, VX und VY. Die Werte in der Spalte TEMP sind die ersten 12 Zeichen der Zeichenfolgen in Zeile 1-50 von Spalte V1, die Werte von Spalte VX sind die ersten 12 Zeichen der Zeichenfolgen in Zeile 1-50 von Spalte V2 und die Werte von Spalte VY sind die ersten 12 Zeichen der Zeichenfolgen in Zeile 1-50 von Spalte V3.

## Zugehörige Funktionen

Funktion	Beschreibung
STRING_CONCAT	Verkettet zwei Zeichenfolgen aus den angegebenen Datenbereichen.
STRING_LENGTH	Gibt die Länge jeder Zeichenfolge im angegebenen Datenbereich zurück

Funktion	Beschreibung
STRING_SEG	Gibt das Zeichenfolgesegment zwischen zwei angegebenen Indizes zurück
STRING_TAIL	Gibt die letzten n Zeichen jeder Zeichenfolge im angegebenen Datenbereich zurück.

---

## STRING\_LENGTH-Makro

Das STRING\_LENGTH-Makro ist nur in IBM Campaign verfügbar.

### Syntax

STRING\_LENGTH(data)

Parameters

data

ASCII-Zeichenfolgewerte, deren Länge berechnet werden soll. Dabei kann es sich um ASCII-Text in Anführungszeichen, eine Textspalte, einen Zellenbereich mit Text oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

### Beschreibung

STRING\_LENGTH gibt die Länge jedes Zeichenfolgewerts im angegebenen Datenbereich zurück. Dabei wird für jede Eingabespalte eine neue Spalte zurückgegeben, die die Länge der entsprechenden Zeichenfolge enthält.

**Anmerkung:** Wenn STRING\_LENGTH auf Spalten mit numerischen Daten angewendet wird, gibt die Funktion Nullen zurück.

### Beispiele

TEMP = STRING_LENGTH("four")
Erstellt eine neue Spalte TEMP, die den Wert 4 enthält.
TEMP = STRING_LENGTH(4)
Erstellt eine neue Spalte TEMP, die den Wert 0 enthält.
TEMP = STRING_LENGTH(V1)
Erstellt eine neue Spalte TEMP, in der jeder Wert die Länge der Zeichenfolge in der entsprechenden Zeile von Spalte V1 darstellt.
TEMP = STRING_LENGTH(V1:V3)
Erstellt drei neue Spalten TEMP, VX und VY. Die Werte in der Spalte TEMP sind die Längen der Zeichenfolgen in den entsprechenden Zeilen von Spalte V1, die Werte von Spalte VX sind die Längen der Zeichenfolgen in den entsprechenden Zeilen von Spalte V2 und die Werte von Spalte VY sind die Längen der Zeichenfolgen in den entsprechenden Zeilen von Spalte V3.

```
TEMP = STRING_LENGTH(V4[1:50]:V6]
```

Erstellt drei neue Spalten TEMP, VX und VY. Die Werte in der Spalte TEMP sind die Längen der Zeichenfolgen in Zeile 1-50 von Spalte V1, die Werte von Spalte VX sind die Längen der Zeichenfolgen in Zeile 1-50 von Spalte V2 und die Werte von Spalte VY sind die Längen der Zeichenfolgen in Zeile 1-50 von Spalte V3.

## Zugehörige Funktionen

Funktion	Beschreibung
STRING_CONCAT	Verkettet zwei Zeichenfolgen aus den angegebenen Datenbereichen.
STRING_HEAD	Gibt die ersten n Zeichen jeder Zeichenfolge im angegebenen Datenbereich zurück.
STRING_SEG	Gibt das Zeichenfolgesegment zwischen zwei angegebenen Indizes zurück
STRING_TAIL	Gibt die letzten n Zeichen jeder Zeichenfolge im angegebenen Datenbereich zurück.

---

## STRING\_PROPER-Makro

Das STRING\_PROPER-Makro Makro ist nur in IBM Campaign verfügbar.

### Syntax

```
STRING_PROPER(data)
```

### Parameter

data

Der Zeichenfolgewart, der konvertiert werden soll.

### Beschreibung

STRING\_PROPER konvertiert jeden Zeichenfolgewart im angegebenen Datenbereich so, dass der erste Buchstabe oder irgendein Buchstabe, der auf ein Leerzeichen oder Symbol (außer dem Unterstrich) folgt, in einen Großbuchstaben und alle anderen Zeichen in Kleinbuchstaben geändert werden. STRING\_PROPER gibt für jede Eingabespalte eine neue Spalte zurück, die die konvertierte Zeichenfolge der entsprechenden Eingabespalte enthält.

### Beispiele

```
Temp = STRING_PROPER
```

---

## STRING\_SEG-Makro

Das STRING\_SEG-Makro Makro ist nur in IBM Campaign verfügbar.

### Syntax

```
STRING_SEG(from, to, data)
```

## Parameter

from

Die Zeichenzahl ab dem Anfang der Zeichenfolge, bei der die Extraktion des Zeichenfolgesegments beginnen soll. Es muss sich um eine positive Ganzzahl größer als null und kleiner als to handeln; andernfalls gibt STRING\_SEG eine leere Zeichenfolge zurück.

to

Die Zeichenzahl ab dem Anfang der Zeichenfolge, bei der die Extraktion des Zeichenfolgesegments enden soll. Es muss sich um eine positive Ganzzahl größer-gleich from handeln. Wenn to gleich from ist (und to kleiner-gleich der Länge der Zeichenfolge ist), wird ein Zeichen zurückgegeben.

data

ASCII-Zeichenfolgewerte. Dabei kann es sich um ASCII-Text in Anführungszeichen, eine Textspalte, einen Zellenbereich mit Text oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

## Beschreibung

STRING\_SEG gibt aus jedem Zeichenfolgewert im angegebenen Datenbereich das Zeichenfolgesegment zwischen zwei Indizes zurück. Wenn from größer als die Länge einer Zeichenfolge ist, wird nichts zurückgegeben. Wenn to größer als die Länge einer Zeichenfolge ist, werden alle Zeichen ab from zurückgegeben.

## Beispiele

<pre>TEMP = STRING_SEG(1, 6, "JAN 15, 1997")</pre>
Erstellt eine neue Spalte TEMP, die die ASCII-Zeichenfolge "JAN 15" enthält.
<pre>TEMP = STRING_SEG(5, 20, "Pressure")</pre>
Erstellt eine neue Spalte TEMP, die die ASCII-Zeichenfolge "sure" enthält.
<pre>TEMP = STRING_SEG(5, 6, V1)</pre>
Erstellt eine neue Spalte TEMP, die das fünfte und das sechste Zeichen jeder Zeichenfolge in der Spalte V1 enthält.
<pre>TEMP = STRING_SEG(10, 20, V1:V3)</pre>
Erstellt drei neue Spalten TEMP, VX und VY. Die Werte in der Spalte TEMP sind die Zeichen 10-20 der Zeichenfolgen in den entsprechenden Zeilen von Spalte V1, die Werte von Spalte VX sind die Zeichen 10-20 der Zeichenfolgen in den entsprechenden Zeilen von Spalte V2 und die Werte von Spalte VY sind die Zeichen 10-20 der Zeichenfolgen in den entsprechenden Zeilen von Spalte V3.
<pre>TEMP = STRING_SEG(5, 10, V4[1:50]:V6]</pre>
Erstellt drei neue Spalten TEMP, VX und VY. Die Werte in der Spalte TEMP sind die Zeichen 5-10 der Zeichenfolgen in Zeile 1-50 von Spalte V1, die Werte von Spalte VX sind die Zeichen 5-10 der Zeichenfolgen in Zeile 1-50 von Spalte V2 und die Werte von Spalte VY sind die Zeichen 5-10 der Zeichenfolgen in Zeile 1-50 von Spalte V3.

## Zugehörige Funktionen

Funktion	Beschreibung
STRING_CONCAT	Verkettet zwei Zeichenfolgen aus den angegebenen Datenbereichen.
STRING_HEAD	Gibt die ersten n Zeichen jeder Zeichenfolge im angegebenen Datenbereich zurück.
STRING_LENGTH	Gibt die Länge jeder Zeichenfolge im angegebenen Datenbereich zurück
STRING_TAIL	Gibt die letzten n Zeichen jeder Zeichenfolge im angegebenen Datenbereich zurück.

---

## STRING\_TAIL-Makro

Das STRING\_TAIL-Makro Makro ist nur in IBM Campaign verfügbar.

### Syntax

```
STRING_TAIL(num_chars, data)
```

### Parameter

num\_chars

Die Anzahl der Zeichen, die vom Ende jeder Zeichenfolge in data zurückgegeben werden sollen. Dieser Wert muss eine positive Ganzzahl größer als null sein.

data

ASCII-Zeichenfolgewerte. Dabei kann es sich um ASCII-Text in Anführungszeichen, eine Textspalte, einen Zellenbereich mit Text oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

### Beschreibung

STRING\_TAIL gibt die letzten num\_chars Zeichen jedes Zeichenfolgewerts im angegebenen Datenbereich zurück. Alle Zeichenfolgewerte werden bis zur Länge der längsten Zeichenfolge mit Nullzeichen "\0" aufgefüllt. Anschließend werden die letzten num\_chars Zeichen jeder Zeichenfolge zurückgegeben. Wenn num\_chars größer als die Anzahl der Zeichen in einer Zeichenfolge ist, wird die gesamte Zeichenfolge zurückgegeben.

### Beispiele

TEMP = STRING_TAIL(3, "JAN 15, 1997")
Erstellt eine neue Spalte TEMP, die die ASCII-Zeichenfolge "997" enthält.
TEMP = STRING_TAIL(10, "Pressure")
Erstellt eine neue Spalte TEMP, die die ASCII-Zeichenfolge "Pressure" enthält.

TEMP = STRING_TAIL(5, V1)
Erstellt eine neue Spalte TEMP, die die letzten fünf Zeichen jeder Zeichenfolge in der Spalte V1 enthält.
TEMP = STRING_TAIL(1, V1:V3)
Erstellt drei neue Spalten TEMP, VX und VY. Die Werte in der Spalte TEMP sind jeweils das letzte Zeichen der Zeichenfolgen in den entsprechenden Zeilen von Spalte V1, die Werte von Spalte VX sind das letzte Zeichen der Zeichenfolgen in den entsprechenden Zeilen von Spalte V2 und die Werte von Spalte VY sind das letzte Zeichen der Zeichenfolgen in den entsprechenden Zeilen von Spalte V3.
TEMP = STRING_TAIL(12, V4[1:50]:V6]
Erstellt drei neue Spalten TEMP, VX und VY. Die Werte in der Spalte TEMP sind die letzten 12 Zeichen der Zeichenfolgen in Zeile 1-50 von Spalte V1, die Werte von Spalte VX sind die letzten 12 Zeichen der Zeichenfolgen in Zeile 1-50 von Spalte V2 und die Werte von Spalte VY sind die letzten 12 Zeichen der Zeichenfolgen in Zeile 1-50 von Spalte V3.

## Zugehörige Funktionen

Funktion	Beschreibung
STRING_CONCAT	Verkettet zwei Zeichenfolgen aus den angegebenen Datenbereichen.
STRING_HEAD	Gibt die ersten n Zeichen jeder Zeichenfolge im angegebenen Datenbereich zurück.
STRING_LENGTH	Gibt die Länge jeder Zeichenfolge im angegebenen Datenbereich zurück
STRING_SEG	Gibt das Zeichenfolgesegment zwischen zwei angegebenen Indizes zurück

---

## SUBSTR- oder SUBSTRING-Makro

Das SUBSTR- oder SUBSTRING-Makro Makro ist in IBM Campaign und IBM Interact verfügbar.

### Syntax

SUBSTR(string\_value, start\_pos[, nchars]) or SUBSTR(string\_value FROM start\_pos[ FOR nchars]) SUBSTRING(string\_value, start\_pos[, nchars]) or SUBSTRING(string\_value FROM start\_pos[ FOR nchars])

### Parameter

string\_value

Die Zeichenfolge, der eine Unterzeichenfolge entnommen werden soll.

start\_pos

Das erste Zeichen jeder Unterzeichenfolge wird extrahiert.

nchars

Die Anzahl der zu extrahierenden Zeichen (muss größer-gleich 0 sein). Wenn dieser Wert nicht angegeben ist, werden alle restlichen Zeichen in `string_value` extrahiert.

## Beschreibung

SUBSTR oder SUBSTRING extrahiert `nchars` Zeichen aus der Zeichenfolge, beginnend bei `start_pos`. Wenn `nchars` nicht angegeben ist, werden durch SUBSTR und SUBSTRING die Zeichen von `start_pos` bis zum Ende der Zeichenfolge extrahiert. Nachfolgende Leerzeichen werden automatisch abgeschnitten. Um Syntaxfehler zu vermeiden, stellen Sie sicher, dass numerische Werte durch ein Komma und ein Leerzeichen getrennt werden, wie in den Beispielen gezeigt.

**Wichtig:** IBM Interact unterstützt nur die folgenden Formate:  
SUBSTR(`string_value`, `start_pos`[, `nchars`]) oder SUBSTRING(`string_value`, `start_pos`[, `nchars`])

## Beispiele

SUBSTR SUBSTR Returns	("abcdef" FROM 1 FOR 2) ("abcdef", 1, 2) 'ab'
SUBSTR SUBSTR Returns	("abcdef" FROM -2 FOR 4) ("abcdef", -2, 4) 'a'
SUBSTR SUBSTR Returns	("abcdef" FROM 3) ("abcdef", 3) 'cdef'

---

## SUM-Makro

Das SUM-Makro Makro ist in IBM Campaign und IBM Interact verfügbar.

### Syntax

SUM(`data` [, `keyword`])

### Parameter

`data`

Die numerischen Werte, deren Summe berechnet werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von `data` finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

`keyword`

Dieses optionale Schlüsselwort legt fest, wie die Berechnung auf den Eingabebereich angewendet wird. Wählen Sie eine der folgenden Angaben aus:

ALL - Wendet die Berechnung auf alle Zellen in `data` an (Standard)

COL - Führt die Berechnung für jede Spalte von `data` gesondert aus

ROW - Führt die Berechnung für jede Zeile von `data` gesondert aus

Weitere Informationen zur Verwendung von Schlüsselwörtern in IBM Campaign finden Sie in „Formatspezifikationen“ auf Seite 9.

**Anmerkung:** Viele Makrofunktionen setzen die Schlüsselwortparameter {ALL | COL | ROW} ein. Diese Schlüsselwörter gelten nicht für IBM Campaign, da es sich bei den Eingabedaten immer um eine Einzelspalte oder ein Einzelfeld handelt. Das Makro verhält sich immer so, als ob das Schlüsselwort COL angegeben würde. Deshalb brauchen Sie diese Schlüsselwörter nicht anzugeben, wenn Sie IBM Campaign verwenden.

## Beschreibung

SUM berechnet die Summe aller Zellen im angegebenen Datenbereich. SUM gibt eine einzelne Spalte zurück.

**Anmerkung:** SUM ist mit der Makrofunktion TOTAL identisch.

## Beispiele

TEMP = SUM(3)	Erstellt eine neue Spalte TEMP, die den Wert drei enthält.
TEMP = SUM((COLUMN(3, 5, 1)))	
	Erstellt eine neue Spalte TEMP, die den Wert neun enthält.
TEMP = SUM(V1)	
	Erstellt eine neue Spalte TEMP, die einen Einzelwert enthält, der die Summe des Inhalts von Spalte V1 darstellt.
TEMP = SUM(V1:V3)	
	Erstellt eine neue Spalte TEMP, die einen Einzelwert enthält, der die Summe des Inhalts der Spalten V1, V2 und V3 darstellt.
TEMP = SUM(V1[1:5]:V4)	
	Erstellt eine neue Spalte TEMP, die einen Einzelwert enthält, der die Summe der Zellen in Zeile 10-20 von Spalte V1 bis V4 darstellt.
TEMP = SUM(V1:V3, COL)	
	Erstellt drei neue Spalten TEMP, VX und VY. Der Einzelwert in der Spalte TEMP ist die Summe des Inhalts von Spalte V1, der Einzelwert in der Spalte VX ist die Summe des Inhalts von Spalte V2 und der Einzelwert in der Spalte VY ist die Summe des Inhalts von Spalte V3.
TEMP = SUM(V1[1:5]:V3, COL)	
	Erstellt drei neue Spalten TEMP, VX und VY, die jeweils einen Einzelwert enthalten. Der Wert in der Spalte TEMP ist die Summe der Zellen in Zeile 1-5 von Spalte V1, der Wert in der Spalte VX ist die Summe der Zellen in Zeile 1-5 von Spalte V2 und der Wert in der Spalte VY ist die Summe der Zellen in Zeile 1-5 von Spalte V3.
TEMP = SUM(V1:V3, ROW)	
	Erstellt eine neue Spalte TEMP, in der jeder Zelleneintrag die Summe der entsprechenden Zeile in Spalte V1, V2 und V3 darstellt.
TEMP = SUM(V1[1:5]:V3, ROW)	
	Erstellt eine neue Spalte TEMP, in der die Zellen in Zeile 1-5 die Summe der entsprechenden Zeile in Spalte V1 bis V3 enthalten. Die anderen Zellen in TEMP sind leer.

## Zugehörige Funktionen

Funktion	Beschreibung
AVG oder MEAN	Berechnet das arithmetische Mittel oder den Durchschnitt eines Zellenbereichs

---

## TAN-Makro

Das TAN-Makro Makro ist nur in IBM Campaign verfügbar.

### Syntax

TAN(data [, units\_keyword])

### Parameter

data

Die numerischen Werte, deren Tangens berechnet werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

units\_keyword

Dieses optionale Schlüsselwort legt fest, ob die Eingabewerte und Ergebnisse als Grad oder als Radiant interpretiert werden. Wählen Sie eine der folgenden Angaben aus:

RADIAN - Führt die Berechnungen in Radiant aus (Standard)

DEGREE - Führt die Berechnungen in Grad aus

Wird dieser Parameter nicht angegeben, ist der Standard Radiant. (Die Umrechnung von Radiant in Grad erfolgt durch Division durch PI und Multiplikation mit 180.)

Weitere Informationen zur Verwendung von Schlüsselwörtern in IBM Campaign finden Sie in „Formatspezifikationen“ auf Seite 9.

### Beschreibung

TAN berechnet den Tangens der Werte im angegebenen Datenbereich. TAN gibt für jede Eingabespalte eine neue Spalte zurück, die den Tangens der Zahlen in der entsprechenden Eingabespalte enthält.

### Beispiele

TEMP = TAN(PI/4) oder TEMP = TAN(PI/4, 0) oder TEMP = TAN(PI/4, RADIANT)
--

Erstellt eine neue Spalte TEMP, die den Wert eins enthält.
--

TEMP = TAN(V1)
Erstellt eine neue Spalte TEMP, in der jeder Wert den Tangens (in Radiant) des Inhalts von Spalte V1 darstellt.
TEMP = TAN(V1:V3, 1) oder TEMP = TAN(V1:V3, DEGREE)
Erstellt drei neue Spalten TEMP, VX und VY. Die Werte in der Spalte TEMP sind jeweils der Tangens des Inhalts von Spalte V1, die Werte von Spalte VX sind der Tangens des Inhalts von Spalte V2 und die Werte von Spalte VY sind der Tangens des Inhalts von Spalte V3. Alle Werte sind in Grad angegeben.
TEMP = TAN(V1[1:5]:V2)
Erstellt zwei neue Spalten TEMP und VX, die jeweils die Werte in Zeile 1-5 enthalten (die anderen Zellen sind leer). Die Werte in der Spalte TEMP sind der Tangens der entsprechenden Zeilen von Spalte V1, die Werte in der Spalte VX sind der Tangens der entsprechenden Zeilen von Spalte V2. Alle Werte sind in Radiant angegeben.

## Zugehörige Funktionen

Funktion	Beschreibung
ATAN	Berechnet den Arkustangens des Inhalts des angegebenen Datenbereichs.
COS	Berechnet den Kosinus des Inhalts des angegebenen Datenbereichs.
COT	Berechnet den Kotangens des Inhalts des angegebenen Datenbereichs.
SIN	Berechnet den Sinus des Inhalts des angegebenen Datenbereichs.
TANH	Berechnet den Hyperbeltangens des Inhalts des angegebenen Datenbereichs.

---

## TANH-Makro

Das TANH-Makro ist nur in IBM Campaign verfügbar.

### Syntax

TANH(data [, units\_keyword])

### Parameter

data

Die numerischen Werte, deren Hyperbeltangens berechnet werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

units\_keyword

Dieses optionale Schlüsselwort legt fest, ob die Eingabewerte und Ergebnisse als Grad oder als Radiant interpretiert werden. Wählen Sie eine der folgenden Angaben aus:

RADIAN - Führt die Berechnungen in Radiant aus (Standard)

DEGREE - Führt die Berechnungen in Grad aus

Wird dieser Parameter nicht angegeben, ist der Standard Radiant. (Die Umrechnung von Radiant in Grad erfolgt durch Division durch PI und Multiplikation mit 180.)

Weitere Informationen zur Verwendung von Schlüsselwörtern in IBM Campaign finden Sie in „Formatspezifikationen“ auf Seite 9.

## Beschreibung

TANH berechnet den Hyperbeltangens der Werte im angegebenen Datenbereich. TANH gibt für jede Eingabespalte eine neue Spalte zurück, die den Hyperbeltangens der Zahlen in der entsprechenden Eingabespalte enthält. Der Hyperbeltangens einer Zahl wird wie folgt berechnet:

$$\tanh(x) = \frac{\sinh(x)}{\cosh(x)}$$

**Anmerkung:** Wenn der Wert  $x$  zu groß ist, wird ein Überlauffehler zurückgegeben. Dies geschieht, wenn  $\tanh(x)$  den maximalen 32-Bit-Gleitkommawert überschreitet. Wenn  $\cosh(x)$  null ist, gibt TANH den maximalen 32-Bit-Gleitkommawert zurück.

## Beispiele

TEMP = TANH(PI) oder TEMP = TANH(PI, 0) oder TEMP = TANH(PI, RADIAN)
Erstellt eine neue Spalte TEMP, die den Wert eins enthält.
TEMP = TANH(V1)
Erstellt eine neue Spalte TEMP, in der jeder Wert den Hyperbeltangens (in Radiant) des Inhalts von Spalte V1 darstellt.
TEMP = TANH(V1:V3, 1) oder TEMP = TANH(V1:V3, DEGREE)
Erstellt drei neue Spalten TEMP, VX und VY. Die Werte in der Spalte TEMP sind jeweils der Hyperbeltangens des Inhalts von Spalte V1, die Werte von Spalte VX sind der Hyperbeltangens des Inhalts von Spalte V2 und die Werte von Spalte VY sind der Hyperbeltangens des Inhalts von Spalte V3. Alle Werte sind in Grad angegeben.
TEMP = TANH(V1[1:5]:V2)
Erstellt zwei neue Spalten TEMP und VX, die jeweils die Werte in Zeile 1-5 enthalten (die anderen Zellen sind leer). Die Werte in der Spalte TEMP sind der Hyperbeltangens der entsprechenden Zeilen von Spalte V1, die Werte in der Spalte VX sind der Hyperbeltangens der entsprechenden Zeilen von Spalte V2. Alle Werte sind in Radiant angegeben.

## Zugehörige Funktionen

Funktion	Beschreibung
ATAN	Berechnet den Arkustangens des Inhalts des angegebenen Datenbereichs.
COSH	Berechnet den Hyperbelkosinus des Inhalts des angegebenen Datenbereichs.

Funktion	Beschreibung
COT	Berechnet den Kotangens des Inhalts des angegebenen Datenbereichs.
SINH	Berechnet den Hyperbelsinus des Inhalts des angegebenen Datenbereichs.
TAN	Berechnet den Tangens des Inhalts des angegebenen Datenbereichs.

---

## TOTAL-Makro

Das TOTAL-Makro Makro ist in IBM Campaign und IBM Interact verfügbar.

### Syntax

TOTAL(data [, keyword])

### Parameter

data

Die numerischen Werte, deren Summe berechnet werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

keyword

Dieses optionale Schlüsselwort legt fest, wie die Berechnung auf den Eingabedatenbereich angewendet wird. Wählen Sie eine der folgenden Angaben aus:

ALL - Wendet die Berechnung auf alle Zellen in data an (Standard)

COL - Führt die Berechnung für jede Spalte von data gesondert aus

ROW - Führt die Berechnung für jede Zeile von data gesondert aus

Weitere Informationen zur Verwendung von Schlüsselwörtern in IBM Campaign finden Sie in „Formatspezifikationen“ auf Seite 9.

**Anmerkung:** Viele Makrofunktionen setzen die Schlüsselwortparameter {ALL | COL | ROW} ein. Diese Schlüsselwörter gelten nicht für IBM Campaign, da es sich bei den Eingabedaten immer um eine Einzelspalte oder ein Einzelfeld handelt. Das Makro verhält sich immer so, als ob das Schlüsselwort COL angegeben würde. Deshalb brauchen Sie diese Schlüsselwörter nicht anzugeben, wenn Sie IBM Campaign verwenden.

### Beschreibung

TOTAL berechnet die Summe aller Zellen im angegebenen Datenbereich.

**Anmerkung:** TOTAL ist mit der Makrofunktion SUM identisch.

## Beispiele

TEMP = TOTAL(3)	Erstellt eine neue Spalte TEMP, die den Wert drei enthält.
TEMP = TOTAL((COLUMN(3, 5, 1)))	
	Erstellt eine neue Spalte TEMP, die den Wert neun enthält.
TEMP = TOTAL(V1)	
	Erstellt eine neue Spalte TEMP, die einen Einzelwert enthält, der die Summe des Inhalts von Spalte V1 darstellt.
TEMP = TOTAL(V1:V3)	
	Erstellt eine neue Spalte TEMP, die einen Einzelwert enthält, der die Summe des Inhalts der Spalten V1, V2 und V3 darstellt.
TEMP = TOTAL(V1[1:5]:V4)	
	Erstellt eine neue Spalte TEMP, die einen Einzelwert enthält, der die Summe der Zellen in Zeile 10-20 von Spalte V1 bis V4 darstellt.
TEMP = TOTAL(V1:V3, COL)	
	Erstellt drei neue Spalten TEMP, VX und VY. Der Einzelwert in der Spalte TEMP ist die Summe des Inhalts von Spalte V1, der Einzelwert in der Spalte VX ist die Summe des Inhalts von Spalte V2 und der Einzelwert in der Spalte VY ist die Summe des Inhalts von Spalte V3.
TEMP = TOTAL(V1[1:5]:V3, COL)	
	Erstellt drei neue Spalten TEMP, VX und VY, die jeweils einen Einzelwert enthalten. Der Wert in der Spalte TEMP ist die Summe der Zellen in Zeile 1-5 von Spalte V1, der Wert in der Spalte VX ist die Summe der Zellen in Zeile 1-5 von Spalte V2 und der Wert in der Spalte VY ist die Summe der Zellen in Zeile 1-5 von Spalte V3.
TEMP = TOTAL(V1:V3, ROW)	
	Erstellt eine neue Spalte TEMP, in der jeder Zelleneintrag die Summe der entsprechenden Zeile in Spalte V1, V2 und V3 darstellt.
TEMP = TOTAL(V1[1:5]:V3, ROW)	
	Erstellt eine neue Spalte TEMP, in der die Zellen in Zeile 1-5 die Summe der entsprechenden Zeile in Spalte V1 bis V3 enthalten. Die anderen Zellen in TEMP sind leer.

## Zugehörige Funktionen

Funktion	Beschreibung
AVG oder MEAN	Berechnet das arithmetische Mittel oder den Durchschnitt eines Zellenbereichs

---

## TRUNCATE-Makro

Das TRUNCATE-Makro Makro ist nur in IBM Campaign verfügbar.

### Syntax

TRUNCATE(data)

## Parameter

data

Die numerischen Werte, die abgeschnitten werden sollen. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

## Beschreibung

TRUNCATE berechnet den Ganzzahlanteil jedes Werts im angegebenen Datenbereich. Dabei wird für jede Eingabespalte eine neue Spalte zurückgegeben, die den ganzzahligen Anteil (den Vorkommaanteil) der Zahlen in der entsprechenden Eingabespalte enthält.

**Anmerkung:** Die Makrofunktionen FRACTION und TRUNCATE sind komplementär, da sie in der Summe die ursprünglichen Werte ergeben.

## Beispiele

TEMP = TRUNCATE(4.3)
Erstellt eine neue Spalte TEMP, die den Wert 4 enthält.
TEMP = TRUNCATE(2.9)
Erstellt eine neue Spalte TEMP, die den Wert -2 enthält.
TEMP = TRUNCATE(V1)
Erstellt eine neue Spalte TEMP, in der jeder Wert die Nachkommastellen des Inhalts von Spalte V1 darstellt.
TEMP = TRUNCATE(V1:V3)
Erstellt drei neue Spalten TEMP, VX und VY. Die Werte in der Spalte TEMP sind die abgeschnittenen Teile von Spalte V1, die Werte von Spalte VX sind die abgeschnittenen Teile von Spalte V2 und die Werte von Spalte VY sind die abgeschnittenen Teile von Spalte V3.
TEMP = TRUNCATE(V1[10:20])
Erstellt eine neue Spalte TEMP, in der die ersten 11 Zellen jeweils den Ganzzahlanteil der Werte in Zeile 10-20 von Spalte V1 enthalten. Die anderen Zellen in TEMP sind leer.
TEMP = TRUNCATE(V1[50:99]:V2)
Erstellt zwei neue Spalten TEMP und VX, die jeweils Werte in Zeile 1-50 enthalten (die anderen Zellen sind leer). Die Werte in der Spalte TEMP sind jeweils der Ganzzahlanteil der Zeilen von Spalte V1, die Werte in der Spalte VX sind der Ganzzahlanteil der Zeilen von Spalte V2.

## Zugehörige Funktionen

Funktion	Beschreibung
CEILING	Berechnet die Obergrenze jedes Werts im angegebenen Datenbereich.
FLOOR	Berechnet die Untergrenze jedes Werts im angegebenen Datenbereich.

Funktion	Beschreibung
FRACTION	Berechnet die Nachkommastellen jedes Werts im angegebenen Datenbereich.

---

## UPPER-Makro

Das UPPER-Makro Makro ist in IBM Campaign und IBM Interact verfügbar.

### Syntax

UPPER(data)

### Parameter

data

Der Zeichenfolgewart, der in Großbuchstaben konvertiert werden soll.

### Beschreibung

UPPER konvertiert jeden Zeichenfolgewart im angegebenen Datenbereich in Großbuchstaben. UPPER gibt für jede Eingabespalte eine neue Spalte zurück, die die in Großbuchstaben konvertierte Zeichenfolge der entsprechenden Eingabespalte enthält.

### Beispiele

Temp = UPPER "gold"	Erstellt eine neue Spalte TEMP, die den Wert "GOLD" enthält.
TEMP = UPPER( "jan 15, 1997")	Erstellt eine neue Spalte TEMP, die die ASCII-Zeichenfolge "JAN 15, 1997" enthält.
TEMP = UPPER( "Pressure")	Erstellt eine neue Spalte TEMP, die die ASCII-Zeichenfolge "PRESSURE" enthält.
TEMP = UPPER(V1)	Erstellt eine neue Spalte TEMP, die die Großbuchstaben jeder Zeichenfolge in der Spalte V1 enthält.

---

## VARIANCE-Makro

Das VARIANCE-Makro Makro ist nur in IBM Campaign verfügbar.

### Syntax

VARIANCE(data [, keyword])

### Parameter

data

Die numerischen Werte, deren Varianz berechnet werden soll. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

keyword

Dieses optionale Schlüsselwort legt fest, wie die Berechnung auf den Eingabedatenbereich angewendet wird. Wählen Sie eine der folgenden Angaben aus:

ALL - Wendet die Berechnung auf alle Zellen in data an (Standard)

COL - Führt die Berechnung für jede Spalte von data gesondert aus

ROW - Führt die Berechnung für jede Zeile von data gesondert aus

Weitere Informationen zur Verwendung von Schlüsselwörtern in IBM Campaign finden Sie in „Formatspezifikationen“ auf Seite 9.

**Anmerkung:** Viele Makrofunktionen setzen die Schlüsselwortparameter {ALL | COL | ROW} ein. Diese Schlüsselwörter gelten nicht für **IBM Campaign**, da es sich bei den Eingabedaten immer um eine Einzelspalte oder ein Einzelfeld handelt. Das Makro verhält sich immer so, als ob das Schlüsselwort COL angegeben würde. Daher brauchen Sie diese Schlüsselwörter nicht anzugeben, wenn Sie **IBM Campaign** verwenden.

## Beschreibung

VARIANCE berechnet die Varianz aller Werte im angegebenen Datenbereich. Varianz ist die Standardabweichung im Quadrat. Die Varianz wird wie folgt berechnet:

$$\frac{1}{n-1} \sum_{j=1}^n (x_j - \text{mean})^2$$

Dabei steht  $x$  für eine Stichprobe,  $n$  für die Anzahl der Stichproben und *Mittelwert* für den Durchschnitt der Verteilung.

**Anmerkung:** Wenn die Anzahl der Stichproben  $n = 1$  ist, gibt VARIANCE einen Fehler zurück.

## Beispiele

```
TEMP = VARIANCE(V1)
```

Erstellt eine neue Spalte TEMP, die einen Einzelwert enthält, der die Varianz des Inhalts von Spalte V1 darstellt.

```
TEMP = VARIANCE(V1:V3)
```

Erstellt eine neue Spalte TEMP, die einen Einzelwert enthält, der die Varianz des Inhalts der Spalten V1, V2 und V3 darstellt.

TEMP = VARIANCE(V1[10:20])
Erstellt eine neue Spalte TEMP, die einen Einzelwert enthält, der die Varianz der Zellen in Zeile 10-20 von Spalte V1 darstellt.
TEMP = VARIANCE(V1[1:5]:V4)
Erstellt eine neue Spalte TEMP, die einen Einzelwert enthält, der die Varianz der Zellen in Zeile 1-5 von Spalte V1 bis V4 darstellt.
TEMP = VARIANCE(V1:V3, COL)
Erstellt drei neue Spalten TEMP, VX und VY. Der Einzelwert in der Spalte TEMP ist die Varianz des Inhalts von Spalte V1, der Einzelwert in der Spalte VX ist die Varianz des Inhalts von Spalte V2 und der Einzelwert in der Spalte VY ist die Varianz des Inhalts von Spalte V3.
TEMP = VARIANCE(V1[1:5]:V3, COL) oder TEMP = VARIANCE(V1[1:5]:V3[1:5], COL)
Erstellt drei neue Spalten TEMP, VX und VY, die jeweils einen Einzelwert enthalten. Der Wert in der Spalte TEMP ist die Varianz der Zellen in Zeile 1-5 von Spalte V1, der Wert in der Spalte VX ist die Varianz der Zellen in Zeile 1-5 von Spalte V2 und der Wert in der Spalte VY ist die Varianz der Zellen in Zeile 1-5 von Spalte V3.
TEMP = VARIANCE(V1:V3, ROW)
Erstellt eine neue Spalte TEMP, in der jeder Zelleneintrag die Varianz der entsprechenden Zeile in Spalte V1, V2 und V3 darstellt.
TEMP = VARIANCE(V1[1:5]:V3, ROW) oder TEMP = VARIANCE(V1[1:5]:V3[1:5], ROW)
Erstellt eine neue Spalte TEMP, in der die Zellen in Zeile 1-5 die Varianz der entsprechenden Zeile in Spalte V1 bis V3 enthalten. Die anderen Zellen in TEMP sind leer.

---

## WEEKDAY-Makro

Das WEEKDAY-Makro ist nur in IBM Campaign verfügbar.

### Syntax

WEEKDAY(data [, conversion\_keyword])

### Parameter

data

Die ASCII-Datumsangaben, die in numerische Werte konvertiert werden sollen, die Wochentage darstellen (1-7). Dabei kann es sich um ASCII-Text in Anführungszeichen, eine Textspalte, einen Zellenbereich mit Text oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

conversion\_keyword

Dieses optionale Schlüsselwort legt fest, wie Textformate für Datums- und Uhrzeitangaben interpretiert werden sollen. Wählen Sie eine der folgenden Angaben aus:

1 - mm/dd/yy (Standard)

2 - dd-mmm-yy

3 - mm/dd/yy hh:mm

Wird dieser Parameter nicht angegeben, ist der Standard 1.

## Beschreibung

WEEKDAY konvertiert Textwerte im angegebenen Datenbereich anhand des angegebenen Umwandlungsformats für Datums- und Uhrzeitangaben in numerische Werte, die Wochentage darstellen. Die Zahl 0 steht für Sonntag, die 1 für Montag usw. bis zur 6 für Samstag. Wenn eine Zeichenfolge mit dem angegebenen `conversion_keyword` nicht geparkt werden kann, gibt WEEKDAY einen Fehler zurück.

## Beispiele

<pre>TEMP = WEEKDAY("1/1/95")</pre>
Erstellt eine neue Spalte TEMP, die die Zahl 0 enthält (der 1. Januar 1995 ist ein Sonntag).
<pre>TEMP = WEEKDAY(V1, 2)</pre>
Erstellt eine neue Spalte TEMP, die für die Zeichenfolgen in Spalte V1 Zahlen für die Wochentage enthält. Für alle Zeichenfolgen in Spalte V1 wird das Format dd-mmm-yy erwartet (andernfalls wird jeweils ??? zurückgegeben).
<pre>TEMP = WEEKDAY(V1:V3, 3)</pre>
Erstellt drei neue Spalten TEMP, VX und VY. Die Spalte TEMP enthält Zahlen, die die Wochentage der Zeichenfolgen in Spalte V1 darstellen. Die Spalte VX enthält Zahlen, die die Wochentage der Zeichenfolgen in Spalte V2 darstellen. Die Spalte VY enthält Zahlen, die die Wochentage der Zeichenfolgen in Spalte V3 darstellen. Für alle Zeichenfolgen in den Spalten V1 bis V3 wird das Format mm/dd/yy hh:mm erwartet (andernfalls wird jeweils ??? zurückgegeben).
<pre>TEMP = WEEKDAY(V1[10:20]:V2, 10)</pre>
Erstellt zwei neue Spalten TEMP und VX. Die Spalte TEMP enthält Zahlen, die die Wochentage der Zeichenfolgen in Zeile 1-20 von Spalte V1 darstellen. Die Spalte VX enthält Zahlen, die die Wochentage der Zeichenfolgen in Zeile 1-20 von Spalte V2 darstellen. Für alle Zeichenfolgen wird das Format mm/dd/yy erwartet (andernfalls wird jeweils ??? zurückgegeben).

## Zugehörige Funktionen

Funktion	Beschreibung
NUMBER	Konvertiert ASCII-Zeichenfolgen für Uhrzeit- und Datumsangaben in numerische Werte

---

## WEEKDAYOF-Makro

Das WEEKDAYOF-Makro ist nur in IBM Campaign verfügbar.

### Syntax

```
WEEKDAYOF(date_string [, input_format])
```

## Parameter

date\_string

Ein Text, der ein gültiges Datum darstellt.

input\_format

Eines der Schlüsselwörter in der folgenden Tabelle, das das Datumsformat von date\_string angibt.

## Beschreibung

WEEKDAYOF gibt für das durch date\_string angegebene Datum den Wochentag als Zahl zwischen 0 und 6 zurück (Sonntag 0, Montag 1 usw.). Wenn input\_format nicht angegeben ist, wird das Standardschlüsselwort DELIM\_M\_D\_Y verwendet.

## Beispiele

WEEKDAYOF("08312000", MMDDYYYY) gibt die Zahl 4 zurück, da der Donnerstag als 4. Wochentag behandelt wird.

**Anmerkung:** Weitere Informationen zu gültigen Datumsformaten finden Sie unter „DATE“ auf Seite 52.

## Zugehörige Funktionen

Funktion	Beschreibung
DAYOF	Gibt den Tag des Monats als Zahl zurück.
MONTHOF	Gibt den Monat des Jahres als Zahl zurück.
YEAROF	Gibt das Jahr als Zahl zurück.

---

## XOR-Makro

Das XOR-Makro ist in IBM Campaign und IBM Interact verfügbar.

### Syntax

```
data1 XOR data2
```

### Parameter

data1

Die nicht negativen Ganzzahlen, die durch bitweises exklusives Oder mit den Werten in data2 verknüpft werden sollen. Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

data2

Die nicht negative(n) Ganzzahl(en), die durch bitweises exklusives Oder mit den Werten in data1 verknüpft werden soll(en). Dabei kann es sich um einen konstanten Wert, eine Spalte, einen Zellenbereich oder einen Ausdruck handeln, dessen Auswertung einen dieser Typen ergibt. Die Anzahl der Spalten in data2 muss mit der Anzahl der Spalten in data1 übereinstimmen, es sei denn, bei data2 handelt es sich um eine Konstante. Informationen zur Formatdefinition von data finden Sie im Abschnitt "Makrofunktionsparameter" im Kapitel des vorliegenden Handbuchs für Ihr IBM Produkt.

## Beschreibung

XOR berechnet das bitweise exklusive Oder zwischen den beiden angegebenen Datenbereichen. XOR gibt für jede Eingabespalte eine neue Spalte zurück, wobei jeweils die entsprechende Spalte von data1 durch bitweises exklusives Oder mit der entsprechenden Spalte von data2 verknüpft wird (d. h., die erste Spalte von data1 wird durch bitweises exklusives Oder mit der ersten Spalte von data2 verknüpft, die zweite Spalte mit der zweiten Spalte usw.).

Wenn es sich bei data2 um eine Konstante handelt, wird jeder Wert in data1 durch bitweises exklusives Oder mit dem betreffenden Wert verknüpft. Wenn data2 eine oder mehrere Spalten enthält, werden die Berechnungen zeilenweise zwischen einer Spalte von data1 und einer Spalte von data2 ausgeführt. Die erste Zeile von data1 wird durch bitweises exklusives Oder mit der ersten Zeile von data2 verknüpft, die zweite Zeile mit der zweiten Zeile usw. Diese zeilenweise ausgeführte Berechnung erzeugt für jede Zeile ein Ergebnis, bis zum letzten Wert der kürzesten Spalte.

**Anmerkung:** Bei dieser Makrofunktion ist die Genauigkeit auf Ganzzahlwerte kleiner als  $2^{24}$  begrenzt. Negative Werte sind nicht zulässig.

## Beispiele

TEMP = 3 XOR 7
Erstellt eine neue Spalte TEMP, die den Wert vier enthält (das bitweise exklusive Oder von 011 und 111 ergibt 100).
TEMP = V1 XOR 8
Erstellt eine neue Spalte TEMP, in der der Inhalt von Spalte V1 jeweils durch bitweises exklusives Oder mit dem Binärwert 1000 verknüpft ist.
TEMP = V1 XOR V1
Erstellt eine neue Spalte TEMP, die nur Nullen enthält (jeder Wert, der durch exklusives Oder mit sich selbst verknüpft wird, ergibt null).
TEMP = V1 XOR V2
Erstellt eine neue Spalte TEMP, in der jeder Zeilenwert von Spalte V1 durch bitweises exklusives Oder mit dem entsprechenden Zeilenwert von Spalte V2 verknüpft ist.
TEMP = V1:V3 XOR V4:V6
Erstellt drei neue Spalten TEMP, VX und VY. In der Spalte TEMP sind die Werte von Spalte V1 durch bitweises exklusives Oder mit den entsprechenden Zeilenwerten von Spalte V4 verknüpft. In der Spalte VX sind die Werte von Spalte V2 durch bitweises exklusives Oder mit den Werten von Spalte V5 verknüpft. In der Spalte VY sind die Werte von Spalte V3 durch bitweises exklusives Oder mit den Werten von Spalte V6 verknüpft.

TEMP = V1[10:20] XOR V2 oder TEMP = V1[10:20] XOR V2[1:11]

Erstellt eine neue Spalte TEMP, in deren ersten 11 Zellen die Werte in Zeile 10-20 von Spalte V1 durch bitweises exklusives Oder mit den Werten in Zeile 1-11 von Spalte V2 verknüpft sind. Die anderen Zellen in TEMP sind leer.

## Zugehörige Funktionen

Funktion	Beschreibung
BIT_AND	Berechnet das bitweise Und zwischen den beiden angegebenen Datenbereichen.
BIT_NOT	Berechnet das bitweise Nicht des Inhalts des angegebenen Datenbereichs.
BIT_OR	Berechnet das bitweise Oder zwischen den beiden angegebenen Datenbereichen.

---

## YEAROF-Makro

Das YEAROF-Makro ist nur in IBM Campaign verfügbar.

### Syntax

YEAROF(date\_string [, input\_format])

### Parameter

date\_string

Ein Text, der ein gültiges Datum darstellt.

input\_format

Eines der Schlüsselwörter in der folgenden Tabelle, das das Datumsformat von date\_string angibt.

### Beschreibung

YEAROF gibt für das durch date\_string angegebene Datum das Jahr als Zahl zurück. Wenn input\_format nicht angegeben ist, wird das Standardschlüsselwort DELIM\_M\_D\_Y verwendet.

### Beispiele

YEAROF("31082000", DDMMYYYY) gibt die Zahl 2000 zurück.

Weitere Informationen zu gültigen Datumsformaten finden Sie unter „DATE“ auf Seite 52.

## Zugehörige Funktionen

Funktion	Beschreibung
DAYOF	Gibt den Tag des Monats als Zahl zurück.
MONTHOF	Gibt den Monat des Jahres als Zahl zurück.

<b>Funktion</b>	<b>Beschreibung</b>
WEEKDAYOF	Gibt den Wochentag als Zahl zurück.



---

## Bevor Sie sich an den technischen Support von IBM wenden

Sollte sich ein Problem nicht mithilfe der Dokumentation beheben lassen, können sich die für den Support zuständigen Kontaktpersonen Ihres Unternehmens telefonisch an den technischen Support von IBM wenden. Verwenden Sie diese Anleitungen, damit das Problem effizient und erfolgreich gelöst wird.

Wenn Sie wissen möchten, wer die für den Support zuständige Kontaktperson Ihres Unternehmens ist, wenden Sie sich an Ihren IBM Administrator.

**Anmerkung:** Der technische Support schreibt bzw. erstellt keine API-Skripts. Wenden Sie sich zur Unterstützung bei der Implementierung unserer API-Angebote an IBM Professional Services.

### Zusammenzustellende Informationen

Halten Sie folgende Informationen bereit, wenn Sie sich an den technischen Support von IBM wenden:

- Kurze Beschreibung der Art Ihres Problems
- Detaillierte Fehlermeldungen, die beim Auftreten des Problems angezeigt werden.
- Schritte zum Reproduzieren des Problems
- Entsprechende Protokolldateien, Sitzungsdateien, Konfigurationsdateien und Daten
- Informationen zu Ihrer -Produkt- und -Systemumgebung, die Sie entsprechend der Beschreibung unter "Systeminformationen" abrufen können.

### Systeminformationen

Bei Ihrem Anruf beim technischen Support von IBM werden Sie um verschiedene Informationen gebeten.

Sofern das Problem Sie nicht an der Anmeldung hindert, finden Sie einen Großteil der benötigten Daten auf der Info-Seite. Dort erhalten Sie Informationen zur installierten IBM Anwendung.

Sie können über **Hilfe > Info** (Help > About) auf die Info-Seite zugreifen. Wenn Sie nicht auf die Info-Seite zugreifen können, prüfen Sie, ob im Installationsverzeichnis Ihrer Anwendung die Datei `version.txt` vorhanden ist.

### Kontaktinformationen für den technischen Support von IBM

Wenn Sie sich an den technischen Support von IBM wenden möchten, finden Sie weitere Informationen auf der Website des technischen Supports für IBM Produkte ([http://www.ibm.com/support/entry/portal/open\\_service\\_request](http://www.ibm.com/support/entry/portal/open_service_request)).

**Anmerkung:** Um eine Supportanforderung einzugeben, müssen Sie sich mit einem IBM Account anmelden. Dieser Account muss mit Ihrer IBM Kundennummer verknüpft sein. Weitere Informationen zum Zuordnen Ihres Accounts zu Ihrer IBM Kundennummer erhalten Sie unter **Unterstützungsressourcen > Gültige Softwareunterstützung** im Support Portal.



---

## Bemerkungen

Die vorliegenden Informationen wurden für Produkte und Services entwickelt, die auf dem deutschen Markt angeboten werden.

Möglicherweise bietet IBM die in dieser Dokumentation beschriebenen Produkte, Services oder Funktionen in anderen Ländern nicht an. Informationen über die gegenwärtig im jeweiligen Land verfügbaren Produkte und Services sind beim zuständigen IBM Ansprechpartner erhältlich. Hinweise auf IBM Lizenzprogramme oder andere IBM Produkte bedeuten nicht, dass nur Programme, Produkte oder Services von IBM verwendet werden können. Anstelle der IBM Produkte, Programme oder Services können auch andere, ihnen äquivalente Produkte, Programme oder Services verwendet werden, solange diese keine gewerblichen oder anderen Schutzrechte von IBM verletzen. Die Verantwortung für den Betrieb von Produkten, Programmen und Services anderer Anbieter liegt beim Kunden.

Für die in diesem Handbuch beschriebenen Erzeugnisse und Verfahren kann es IBM Patente oder Patentanmeldungen geben. Mit der Auslieferung dieses Handbuchs ist keine Lizenzierung dieser Patente verbunden. Lizenzanforderungen sind schriftlich an folgende Adresse zu richten (Anfragen an diese Adresse müssen auf Englisch formuliert werden):

IBM Director of Licensing  
IBM Europe, Middle East & Africa  
Tour Descartes  
2, avenue Gambetta  
92066 Paris La Defense  
France

Trotz sorgfältiger Bearbeitung können technische Ungenauigkeiten oder Druckfehler in dieser Veröffentlichung nicht ausgeschlossen werden. Die hier enthaltenen Informationen werden in regelmäßigen Zeitabständen aktualisiert und als Neuausgabe veröffentlicht. IBM kann ohne weitere Mitteilung jederzeit Verbesserungen und/oder Änderungen an den in dieser Veröffentlichung beschriebenen Produkten und/oder Programmen vornehmen.

Verweise in diesen Informationen auf Websites anderer Anbieter werden lediglich als Service für den Kunden bereitgestellt und stellen keinerlei Billigung des Inhalts dieser Websites dar. Das über diese Websites verfügbare Material ist nicht Bestandteil des Materials für dieses IBM Produkt. Die Verwendung dieser Websites geschieht auf eigene Verantwortung.

Werden an IBM Informationen eingesandt, können diese beliebig verwendet werden, ohne dass eine Verpflichtung gegenüber dem Einsender entsteht.

Lizenznehmer des Programms, die Informationen zu diesem Produkt wünschen mit der Zielsetzung: (i) den Austausch von Informationen zwischen unabhängig voneinander erstellten Programmen und anderen Programmen (einschließlich des vorliegenden Programms) sowie (ii) die gemeinsame Nutzung der ausgetauschten Informationen zu ermöglichen, wenden sich an folgende Adresse:

IBM Corporation  
B1WA LKG1

550 King Street  
Littleton, MA 01460-1250  
USA

Die Bereitstellung dieser Informationen kann unter Umständen von bestimmten Bedingungen - in einigen Fällen auch von der Zahlung einer Gebühr - abhängig sein.

Die Lieferung des in diesem Dokument beschriebenen Lizenzprogramms sowie des zugehörigen Lizenzmaterials erfolgt auf der Basis der IBM Rahmenvereinbarung bzw. der Allgemeinen Geschäftsbedingungen von IBM, der IBM Internationalen Nutzungsbedingungen für Programmpakete oder einer äquivalenten Vereinbarung.

Alle in diesem Dokument enthaltenen Leistungsdaten stammen aus einer kontrollierten Umgebung. Die Ergebnisse, die in anderen Betriebsumgebungen erzielt werden, können daher erheblich von den hier erzielten Ergebnissen abweichen. Einige Daten stammen möglicherweise von Systemen, deren Entwicklung noch nicht abgeschlossen ist. Eine Gewährleistung, dass diese Daten auch in allgemein verfügbaren Systemen erzielt werden, kann nicht gegeben werden. Darüber hinaus wurden einige Daten unter Umständen durch Extrapolation berechnet. Die tatsächlichen Ergebnisse können davon abweichen. Benutzer dieses Dokuments sollten die entsprechenden Daten in ihrer spezifischen Umgebung prüfen.

Alle Informationen zu Produkten anderer Anbieter stammen von den Anbietern der aufgeführten Produkte, deren veröffentlichten Ankündigungen oder anderen allgemein verfügbaren Quellen. IBM hat diese Produkte nicht getestet und kann daher keine Aussagen zu Leistung, Kompatibilität oder anderen Merkmalen machen. Fragen zu den Leistungsmerkmalen von Produkten anderer Anbieter sind an den jeweiligen Anbieter zu richten.

Aussagen über Pläne und Absichten von IBM unterliegen Änderungen oder können zurückgenommen werden und repräsentieren nur die Ziele von IBM.

Alle von IBM angegebenen Preise sind empfohlene Richtpreise und können jederzeit ohne weitere Mitteilung geändert werden. Händlerpreise können unter Umständen von den hier genannten Preisen abweichen.

Diese Veröffentlichung enthält Beispiele für Daten und Berichte des alltäglichen Geschäftsablaufs. Sie sollen nur die Funktionen des Lizenzprogramms illustrieren und können Namen von Personen, Firmen, Marken oder Produkten enthalten. Alle diese Namen sind frei erfunden; Ähnlichkeiten mit tatsächlichen Namen und Adressen sind rein zufällig.

#### COPYRIGHTLIZENZ:

Diese Veröffentlichung enthält Beispielanwendungsprogramme, die in Quellsprache geschrieben sind und Programmier Techniken in verschiedenen Betriebsumgebungen veranschaulichen. Sie dürfen diese Beispielprogramme kostenlos kopieren, ändern und verteilen, wenn dies zu dem Zweck geschieht, Anwendungsprogramme zu entwickeln, zu verwenden, zu vermarkten oder zu verteilen, die mit der Anwendungsprogrammierschnittstelle für die Betriebsumgebung konform sind, für die diese Beispielprogramme geschrieben werden. Diese Beispiele wurden nicht unter allen denkbaren Bedingungen getestet. Daher kann IBM die Zuverlässigkeit, Wartungsfreundlichkeit oder Funktion dieser Programme weder zusagen noch gewährleisten. Die Beispielprogramme werden ohne Wartung (auf "as-is"-Basis)

und ohne jegliche Gewährleistung zur Verfügung gestellt. IBM übernimmt keine Haftung für Schäden, die durch die Verwendung der Beispielprogramme entstehen.

Wird dieses Buch als Softcopy (Book) angezeigt, erscheinen keine Fotografien oder Farbabbildungen.

---

## Marken

IBM, das IBM Logo und [ibm.com](http://ibm.com) sind Marken oder eingetragene Marken der IBM Corporation in den USA und/oder anderen Ländern. Weitere Produkt- und Servicennamen können Marken von IBM oder anderen Unternehmen sein. Eine aktuelle Liste der IBM Marken finden Sie auf der Webseite "Copyright and trademark information" unter [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

---

## Hinweise zu Datenschutzrichtlinien und Nutzungsbedingungen

IBM Softwareprodukte, einschließlich Software as a Service-Lösungen ("Softwareangebote"), können Cookies oder andere Technologien verwenden, um Informationen zur Produktnutzung zu erfassen, die Endbenutzererfahrung zu verbessern und Interaktionen mit dem Endbenutzer anzupassen oder zu anderen Zwecken. Ein Cookie ist ein Datenelement, das von einer Website an Ihren Browser gesendet wird und dann als Tag auf Ihrem Computer gespeichert werden kann, mit dem Ihr Computer identifiziert wird. In vielen Fällen werden von diesen Cookies keine personenbezogenen Daten erfasst. Wenn ein Softwareangebot, das von Ihnen verwendet wird, die Erfassung personenbezogener Daten anhand von Cookies und ähnlichen Technologien ermöglicht, werden Sie im Folgenden über die hierbei geltenden Besonderheiten informiert.

Abhängig von den bereitgestellten Konfigurationen kann dieses Softwareangebot Sitzungscookies und permanente Cookies verwenden, mit denen der Benutzername des Benutzers und andere personenbezogene Daten zum Zwecke des Sitzungsmanagements, zur Verbesserung der Benutzerfreundlichkeit und zu anderen funktionsbezogenen Zwecken sowie zur Nutzungsüberwachung erfasst werden. Diese Cookies können deaktiviert werden. Durch die Deaktivierung kann jedoch auch die von ihnen bereitgestellte Funktionalität nicht mehr genutzt werden.

Die Erfassung personenbezogener Daten mithilfe von Cookies und ähnlichen Technologien wird durch verschiedene rechtliche Bestimmungen geregelt. Wenn die für dieses Softwareangebot implementierten Konfigurationen Ihnen als Kunde die Möglichkeit bieten, personenbezogene Daten von Endbenutzern über Cookies und andere Technologien zu erfassen, dann sollten Sie ggf. juristische Beratung zu den geltenden Gesetzen für eine solche Datenerfassung in Anspruch nehmen. Dies gilt auch in Bezug auf die Anforderungen, die vom Gesetzgeber in Bezug auf Hinweise und die Einholung von Einwilligungen vorgeschrieben werden.

IBM setzt voraus, dass Kunden folgende Bedingungen erfüllen: (1) Sie stellen einen klar erkennbaren und auffälligen Link zu den Nutzungsbedingungen der Kundenwebsite (z. B. Datenschutzerklärung) bereit. Dieser Link muss wiederum einen Link zu der Vorgehensweise von IBM und des Kunden bei der Datenerhebung und Datennutzung umfassen. (2) Sie weisen darauf hin, dass Cookies und Clear GIFs/ Web-Beacons von IBM im Auftrag des Kunden auf dem Computer des Besuchers platziert werden. Dieser Hinweis muss eine Erläuterung hinsichtlich des Zwecks dieser Technologie umfassen. (3) Sie müssen in dem gesetzlich vorgeschriebenen Umfang die Einwilligung von Websitebesuchern einholen, bevor Cookies und

Clear GIFs/Web-Beacons vom Kunden oder von IBM im Auftrag des Kunden auf den Geräten der Websitebesucher platziert werden.

Weitere Informationen zur Verwendung verschiedener Technologien einschließlich der Verwendung von Cookies zu diesen Zwecken finden Sie im IBM Online Privacy Statement unter der Webadresse <http://www.ibm.com/privacy/details/us/en> im Abschnitt mit dem Titel "Cookies, Web Beacons and Other Technologies".





Gedruckt in Deutschland