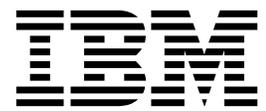


V 10 R 1  
2017 年 10 月

# IBM Macros for IBM Marketing Software 用户指南

The IBM logo, consisting of the letters "IBM" in a bold, sans-serif font, with each letter formed by eight horizontal stripes of varying lengths.

**注意**

在使用本信息及其支持的产品前，请先阅读第 147 页的『声明』中的信息。

# 目录

## 第 1 章 在 IBM Campaign 中使用宏 . . . 1

IBM Campaign 宏函数摘要 . . . . .	1
统计函数. . . . .	1
数学函数和三角函数 . . . . .	2
字符串函数 . . . . .	4
日期和时间函数 . . . . .	5
分组函数. . . . .	6
其他函数. . . . .	6
适用于 IBM Campaign 的宏函数参数 . . . . .	6
格式规范. . . . .	6
使用常量. . . . .	7

## 第 2 章 在 IBM Interact 中使用宏 . . . . 9

IBM Interact 宏函数摘要 . . . . .	9
统计函数. . . . .	9
数学函数和三角函数 . . . . .	10
字符串函数 . . . . .	11
日期和时间函数 . . . . .	11
其他函数 . . . . .	12
适用于 IBM Interact 的宏函数参数 . . . . .	12
格式规范 . . . . .	12
使用常量 . . . . .	12

## 第 3 章 IBM 宏参考 . . . . . 15

有效日期格式的关键字 . . . . .	15
ABS 宏 . . . . .	16
ACOS 宏 . . . . .	17
ACOT 宏 . . . . .	19
ADD_MONTHS 宏 . . . . .	20
AND 宏 . . . . .	21
ASIN 宏 . . . . .	22
ATAN 宏 . . . . .	24
AVG 宏 . . . . .	25
BETWEEN 宏 . . . . .	27
BIT_AND 宏 . . . . .	28
BIT_NOT 宏 . . . . .	29
BIT_OR 宏 . . . . .	30
BIT_XOR 宏 . . . . .	32
CEILING 宏 . . . . .	33
COLUMN 宏 . . . . .	34
COS 宏 . . . . .	35
COSH 宏 . . . . .	37
COT 宏 . . . . .	38
COUNT 宏 . . . . .	40
CURRENT_DATE 宏 . . . . .	41
CURRENT_DAY 宏 . . . . .	42
CURRENT_JULIAN 宏 . . . . .	42
CURRENT_MONTH 宏 . . . . .	43
CURRENT_TIME 宏 . . . . .	43
Web 应用程序的日期设置. . . . .	43
CURRENT_WEEKDAY 宏 . . . . .	45

CURRENT_YEAR 宏 . . . . .	45
DATE . . . . .	46
DATE_FORMAT 宏 . . . . .	47
DATE_JULIAN 宏 . . . . .	49
DATE_STRING 宏 . . . . .	49
DAY_BETWEEN 宏 . . . . .	51
DAY_FROMNOW 宏 . . . . .	52
DAY_INTERVAL 宏 . . . . .	52
DAYOF 宏 . . . . .	53
DISTANCE 宏 . . . . .	54
DIV 宏 . . . . .	55
EQ 宏 . . . . .	56
EXP 宏 . . . . .	58
EXTERNALCALLOUT 宏 . . . . .	59
FACTORIAL 宏 . . . . .	60
FLOOR 宏 . . . . .	61
FORMAT 宏 . . . . .	62
FRACTION 宏 . . . . .	64
GE 宏 . . . . .	66
GROUPBY 宏 . . . . .	67
GROUPBY_WHERE 宏 . . . . .	69
GT 宏 . . . . .	70
IF 宏 . . . . .	72
IN 宏 . . . . .	73
INT 宏 . . . . .	74
INVERSE 宏 . . . . .	75
IS 宏 . . . . .	76
ISERROR 宏 . . . . .	76
ISODD 宏 . . . . .	77
ISEVEN 宏 . . . . .	78
ISODD 宏 . . . . .	79
LE 宏 . . . . .	80
LIKE 宏 . . . . .	82
LN 或 LOG 宏 . . . . .	84
LOG2 宏 . . . . .	85
LOG10 宏 . . . . .	86
LOWER 宏 . . . . .	87
LT 宏 . . . . .	87
LTRIM 宏 . . . . .	89
MAX 宏 . . . . .	89
MEAN 宏 . . . . .	91
MIN 宏 . . . . .	92
MINUS 宏 . . . . .	94
MOD 宏 . . . . .	95
MONTHOF 宏 . . . . .	97
MULT 宏 . . . . .	98
NE 宏 . . . . .	99
NOT 宏 . . . . .	101
NUMBER 宏 . . . . .	102
OR 宏 . . . . .	107
POSITION 宏 . . . . .	108

PLUS 宏 . . . . .	110	SUBSTR 或 SUBSTRING 宏 . . . . .	129
POW 宏 . . . . .	111	SUM 宏 . . . . .	130
RANDOM 宏 . . . . .	113	TAN 宏 . . . . .	132
RANDOM_GAUSS 宏 . . . . .	114	TANH 宏 . . . . .	133
ROUND 宏 . . . . .	115	TOTAL 宏 . . . . .	134
ROWNUM 宏 . . . . .	116	TRUNCATE 宏 . . . . .	136
RTRIM 宏 . . . . .	116	UPPER 宏 . . . . .	137
SIGN 宏 . . . . .	117	VARIANCE 宏 . . . . .	138
SIN 宏 . . . . .	118	WEEKDAY 宏 . . . . .	139
SINH 宏 . . . . .	119	WEEKDAYOF 宏 . . . . .	140
SQRT 宏 . . . . .	120	XOR 宏 . . . . .	141
STDV 或 STDEV 宏 . . . . .	121	YEAROF 宏 . . . . .	142
STRING_CONCAT 宏 . . . . .	123		
STRING_HEAD 宏 . . . . .	124	<b>在与 IBM 技术支持联系之前 . . . . . 145</b>	
STRING_LENGTH 宏 . . . . .	125		
STRING_PROPER 宏 . . . . .	126	<b>声明 . . . . . 147</b>	
STRING_SEG 宏 . . . . .	127	商标 . . . . .	148
STRING_TAIL 宏 . . . . .	128	隐私策略和使用条款注意事项 . . . . .	149

## 第 1 章 在 IBM Campaign 中使用宏

本章提供有关 IBM® Campaign 宏的用法信息。尝试在 IBM Campaign 中使用宏之前，请务必阅读本章。

### IBM Campaign 宏函数摘要

本节中的表按类别概述宏函数。

第 15 页的第 3 章，『IBM 宏参考』按字母顺序提供每个宏函数的详细参考页面。

**要点：**宏可能同时适用于 IBM Campaign 和 IBM Interact，也可能仅适用于其中一款产品。宏描述会指出它们所适用的产品。

有关宏函数输入参数的信息，请参阅第 6 页的『适用于 IBM Campaign 的宏函数参数』。

### 统计函数

宏名称	返回结果	描述
AVG	对于 ALL 关键字：新列，其中包含单个值；对于 COL 关键字：一列，其中针对每个输入列包含单个值；对于 ROW 关键字：一列，其中针对每一行包含一个值。	计算一系列单元格的平均值或算术平均值
COUNT	新列中的单个值。	计算所指定数据范围内值的数量
MAX	对于 ALL 关键字：新列，其中包含单个值；对于 COL 关键字：一列，其中针对每个输入列包含单个值；对于 ROW 关键字：一列，其中针对每一行包含一个值。	计算一系列单元格的极大值
MEAN	对于 ALL 关键字：新列，其中包含单个值；对于 COL 关键字：一列，其中针对每个输入列包含单个值；对于 ROW 关键字：一列，其中针对每一行包含一个值。	计算一系列单元格的平均值或算术平均值
MIN	对于 ALL 关键字：新列，其中包含单个值；对于 COL 关键字：一列，其中针对每个输入列包含单个值；对于 ROW 关键字：一列，其中针对每一行包含一个值。	计算一系列单元格的最小值

宏名称	返回结果	描述
STDV 或 STDEV	对于 ALL 关键字：新列，其中包含单个值；对于 COL 关键字：一列，其中针对每个输入列包含单个值；对于 ROW 关键字：一列，其中针对每一行包含一个值。	计算一系列单元格的标准差
VARIANCE	对于 ALL 关键字：新列，其中包含单个值；对于 COL 关键字：一列，其中针对每个输入列包含单个值；对于 ROW 关键字：一列，其中针对每一行包含一个值。	计算一系列单元格的方差

## 数学函数和三角函数

宏名称	返回结果	描述
ABS	针对每个输入列返回一列	计算所指定数据范围内内容的绝对值
ACOS	针对每个输入列返回一列	计算所指定数据范围内内容的反余弦值
ACOT	针对每个输入列返回一列	计算所指定数据范围内内容的反余切值
ASIN	针对每个输入列返回一列	计算所指定数据范围内内容的反正弦值
ATAN	针对每个输入列返回一列	计算所指定数据范围内内容的反正切值
AVG	针对每个输入列返回一列	计算所指定数据范围内单元格的平均值或算术平均值
BETWEEN	针对每个输入列返回一列	比较两个值，以确定提供的值是否介于另外两个值之间
CEILING	针对每个输入列返回一列	计算所指定数据范围内每个值的向上取整
COLUMN	针对每个输入列返回一列	创建新列，在每一列中垂直并置输入值
COS	针对每个输入列返回一列	计算所指定数据范围内内容的余弦值
COSH	针对每个输入列返回一列	计算所指定数据范围内内容的双曲余弦值
COT	针对每个输入列返回一列	计算所指定数据范围内内容的余切值
COUNT	一个列，其中包含单个值	计算所指定数据范围内包含值的单元格数量
EXP	针对每个输入列返回一列	以自然数 (e) 为底数，并以所指定数据范围中的每个单元格内容为指数，计算乘幂结果

宏名称	返回结果	描述
FACTORIAL	针对每个输入列返回一列	计算所指定数据范围内每个值的阶乘
FLOOR	针对每个输入列返回一列	计算所指定数据范围内每个值的向下取整
FRACTION	针对每个输入列返回一列	返回所指定数据范围内每个值的小数部分
INT	针对每个输入列返回一列	计算所指定数据范围内内容的整数值（向下舍入）
INVERSE	针对每个输入列返回一列	计算所指定数据范围内内容的相反数
LN	针对每个输入列返回一列	计算所指定数据范围内内容的自然对数
LOG	针对每个输入列返回一列	计算所指定数据范围内内容的自然对数
LOG2	针对每个输入列返回一列	计算所指定数据范围内内容以 2 为底的对数
LOG10	针对每个输入列返回一列	计算所指定数据范围内内容以 10 为底的对数
MAX	对于 ALL 关键字：新列，其中包含单个值；对于 COL 关键字：一列，其中针对每个输入列包含单个值；对于 ROW 关键字：一列，其中针对每一行包含一个值。	计算一系列单元格的最高值
MEAN	对于 ALL 关键字：新列，其中包含单个值；对于 COL 关键字：一列，其中针对每个输入列包含单个值；对于 ROW 关键字：一列，其中针对每一行包含一个值。	计算一系列单元格的平均值或算术平均值
MIN	对于 ALL 关键字：新列，其中包含单个值；对于 COL 关键字：一列，其中针对每个输入列包含单个值；对于 ROW 关键字：一列，其中针对每一行包含一个值。	计算一系列单元格的最低值
RANDOM	一个列，其中包含所指定数目的值	返回指定数目的随机数
RANDOM_GAUSS	一个列，其中包含所指定数目的值	根据高斯分布返回指定数目的随机值
ROUND	针对每个输入列返回一列	计算所指定数据范围内内容的取整值
SIGN	针对每个输入列返回一列	计算所指定数据范围内值的符号（正或负）
SIN	针对每个输入列返回一列	计算所指定数据范围内内容的正弦值

宏名称	返回结果	描述
SINH	针对每个输入列返回一列	计算所指定数据范围中内容的双曲正弦值
SQRT	针对每个输入列返回一列	计算所指定数据范围中内容的平方根
STDV 或 STDEV	对于 ALL 关键字：新列，其中包含单个值；对于 COL 关键字：一列，其中针对每个输入列包含单个值；对于 ROW 关键字：一列，其中针对每一行包含一个值。	计算一系列单元格的标准差
SUM	对于 ALL 关键字：新列，其中包含单个值；对于 COL 关键字：一列，其中针对每个输入列包含单个值；对于 ROW 关键字：一列，其中针对每一行包含一个值。	计算某个范围的单元格的总和
TAN	针对每个输入列返回一列	计算所指定数据范围中内容的正切值
TANH	针对每个输入列返回一列	计算所指定数据范围中内容的双曲正切值
TOTAL	对于 ALL 关键字：新列，其中包含单个值；对于 COL 关键字：一列，其中针对每个输入列包含单个值；对于 ROW 关键字：一列，其中针对每一行包含一个值。	计算某个范围的单元格的总和
TRUNCATE	针对每个输入列返回一列	返回所指定数据范围内每个值的非小数部分
VARIANCE	对于 ALL 关键字：新列，其中包含单个值；对于 COL 关键字：一列，其中针对每个输入列包含单个值；对于 ROW 关键字：一列，其中针对每一行包含一个值。	计算一系列单元格的方差

## 字符串函数

宏名称	返回结果	描述
FORMAT	针对每个输入列返回一列	提供对数字和字符串的输出格式控制（例如，输出宽度、对齐、数字精度、小数点符号、分组符号等）。返回已设置格式的输出字符串。
LIKE	针对每个输入列返回一列	确定文本字符串是否与指定的模式匹配
LOWER	针对每个输入列返回一列	将字符串值转换为小写

宏名称	返回结果	描述
LTRIM	针对每个输入列返回一列	除去每个字符串值中的前导空格字符
NUMBER	针对每个输入列返回一列	将表示时间和日期的 ASCII 文本字符串转换为数值
POSITION	针对每个输入列返回一列	返回模式在文本字符串中的起始位置
RTRIM	针对每个输入列返回一列	除去每个字符串值中的尾部空格字符
STRING_CONCAT	一个列，其中针对最短输入列中的每一行包含一个值	将所指定数据范围内的文本字符串并置
STRING_HEAD	针对每个输入列返回一列	返回所指定数据范围内每个字符串的前 $n$ 个字符
STRING_LENGTH	针对每个输入列返回一列	返回所指定数据范围内每个字符串的长度
STRING_PROPER	针对每个输入列返回一列	按以下方式转换每个字符串值：将首字母或者跟在空格字符或符号后的任何非下划线字母转换为大写，并将所有其他字符转换为小写
STRING_SEG	针对每个输入列返回一列	返回两个指定下标之间的字符串段
STRING_TAIL	针对每个输入列返回一列	返回所指定数据范围内每个字符串的最后 $n$ 个字符
SUBSTR 或 SUBSTRING	针对每个输入列返回一列	返回字符串中从起始位置开始的字符
UPPER	针对每个输入列返回一列	将字符串值转换为大写

## 日期和时间函数

宏名称	返回结果	描述
CURRENT_DATE	针对每个输入列返回一列	以 format 格式返回当前日期
CURRENT_DAY	针对每个输入列返回一列	以 1-31 的数字形式返回当天是一个月中的第几天
CURRENT_JULIAN	针对每个输入列返回一列	返回当前日期的儒略数
CURRENT_MONTH	针对每个输入列返回一列	以 1-12 的数字形式返回一年内的当前月份
CURRENT_TIME	针对每个输入列返回一列	以字符串形式返回当前时间
CURRENT_WEEKDAY	针对每个输入列返回一列	以 0-6 的数字形式返回当天是一周中的第几天
CURRENT_YEAR	针对每个输入列返回一列	以数字形式返回当前年份
DATE	针对每个输入列返回一列	将日期字符串转换为儒略日期
DATE_FORMAT	针对每个输入列返回一列	转换日期格式
DATE_JULIAN	针对每个输入列返回一列	返回儒略日期
DATE_STRING	针对每个输入列返回一列	返回儒略日期的日期字符串

宏名称	返回结果	描述
DAY_BETWEEN	针对每个输入列返回一列	返回两个日期之间的天数
DAY_FROMNOW	针对每个输入列返回一列	返回从当前日期到指定日期的天数
DAY_INTERVAL	针对每个输入列返回一列	返回两个日期之间的天数
DAYOF	针对每个输入列返回一列	以数字形式返回一个月中的日子
MONTHOF	针对每个输入列返回一列	以数字形式返回一年中的月份
WEEKDAY	针对每个输入列返回一列	将 ASCII 文本日期字符串转换为一周中的第几天
WEEKDAYOF	针对每个输入列返回一列	以数字形式返回某个日期是一周中的第几天
YEAROF	针对每个输入列返回一列	以数字形式返回年份

## 分组函数

宏名称	返回结果	描述
GROUPBY	一个新列，其中针对每一行包含一个值	对一个组内的多行数据进行汇总
GROUPBY_WHERE	一个新列，其中针对每一行包含一个值	对一个组内符合指定条件的多行数据进行汇总

## 其他函数

宏名称	返回结果	描述
IF	一个列，其中针对最短输入列中的每一行包含一个值	开始条件式 if-then-else 语句
ISERROR	一个列，其中针对最短输入列中的每一行包含一个值	如果输入行中的任何值包含错误 (???) 单元格则返回 1，否则返回 0
ISEVEN	针对每个输入列返回一列	检验输入值是否为偶数（即，可被 2 整除）
ISODD	针对每个输入列返回一列	检验输入值是否为奇数（即，不可被 2 整除）
ROWNUM	针对每个输入列返回一列	生成从 1 到记录数的序号

## 适用于 IBM Campaign 的宏函数参数

本节描述 IBM Campaign 中宏函数的参数和用法。

### 格式规范

本节描述一些常用参数的格式。该格式适用于本章中宏函数规范对这些参数进行的所有引用。

## data

data 参数表示宏函数所要处理的数据列。

此参数可以是常量或字段。请参阅特定的宏函数以了解详细信息。

注： IBM Campaign 不支持同时对多个字段执行计算，也不支持对行的子集执行计算。

某些其他参数名称也使用与 data 相同的格式。这些参数的描述会引用此部分和格式。

## keyword

keyword 参数控制宏函数的行为。它指出可以指定关键字（如果省略此参数，那么将使用缺省关键字）。在每个宏函数中，关键字选项以如下格式列出：

**{choice1 | choice2 | choice3}**

请选择提供所需行为的关键字选项。缺省选项以粗体显示。例如，给定下列选项：

**{RADIANS | DEGREES}**

下列宏函数均有效：

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

注：许多宏函数使用关键字参数 {ALL | COL | ROW}。这些关键字在 IBM Campaign 中不适用，因为输入数据始终为单个列或字段。这个宏的行为始终像是已指定 COL 关键字一样。因此，使用 IBM Campaign 时，无需指定这些关键字。

## 使用常量

大多数宏函数参数可使用数字常量或求值为数字常量的表达式（处理字符串的宏函数可使用字符串常量）。

在逐条记录地执行运算（例如，将两个数字列相加）的宏函数中，使用常量等同于指定一个其中每一行都包含该常量值的列。在本质上，提供常量作为输入参数时，该常量将扩展为长度与输入相同。

一些宏函数可使用 ASCII 文本字符串和数字常量。既可使用数字常量又可使用 ASCII 文本字符串的参数在每个宏函数的“参数”部分中作了说明。

下表中提供了一些示例。

函数定义	如何解释常量
<code>PERCENT_UTILIZ = (CURR_BAL*100)/CREDIT_LIM</code>	常量 100 解释为这样一个列：它包含与列 CURR_BAL 相同的行数，且每一行都包含常量 100。派生字段 PERCENT_UTILIZ 将包含 CURR_BAL 的每个值乘以 100 然后除以 CREDIT_LIM 的每个值所得到的结果。
<code>NAME = STRING_CONCAT ("Mr. ", LAST_NAME)</code>	常量 "Mr." 解释为这样一个列：它包含与列 LAST_NAME 相同的行数，且每一行都包含常量 "Mr."。派生字段 NAME 将包含在 LAST_NAME 中的每个文本字符串前面添加 "Mr." 作为前缀后所得到的结果。



---

## 第 2 章 在 IBM Interact 中使用宏

本章提供有关 IBM Interact 宏的用法信息。尝试在 IBM Interact 中使用宏之前，请务必阅读本章。

### 公式帮助程序和宏表达式语法检查

**要点：**"公式帮助程序"对话框（包括其语法检查功能）目前根据 IBM Campaign 所支持的内容对宏表达式进行验证。然而，IBM Interact 仅支持部分 Campaign 宏功能。因此，您必须确保用于 Interact 的宏和关键字（例如，日期格式关键字）受支持。请在本指南的"宏参考"章节中查找与 Interact 相关的说明。

---

## IBM Interact 宏函数摘要

下列各节中的表提供特定于 IBM Interact 的宏的详细描述。

第 15 页的第 3 章，『IBM 宏参考』按字母顺序提供每个宏函数的详细参考页面。

**要点：**宏可能同时适用于 IBM Campaign 和 IBM Interact，也可能仅适用于其中一款产品。宏描述会指出它们所适用的产品。

第 12 页的『适用于 IBM Interact 的宏函数参数』提供适用于 Interact 的宏函数输入参数的相关信息。

### 统计函数

宏名称	返回结果	描述
AVG	对于 ALL 关键字：新列，其中包含单个值；对于 COL 关键字：一列，其中针对每个输入列包含单个值；对于 ROW 关键字：一列，其中针对每一行包含一个值。	计算一系列单元格的平均值或算术平均值
MAX	对于 ALL 关键字：新列，其中包含单个值；对于 COL 关键字：一列，其中针对每个输入列包含单个值；对于 ROW 关键字：一列，其中针对每一行包含一个值。	计算一系列单元格的最高值
MEAN	对于 ALL 关键字：新列，其中包含单个值；对于 COL 关键字：一列，其中针对每个输入列包含单个值；对于 ROW 关键字：一列，其中针对每一行包含一个值。	计算一系列单元格的平均值或算术平均值

宏名称	返回结果	描述
MIN	对于 ALL 关键字：新列，其中包含单个值；对于 COL 关键字：一列，其中针对每个输入列包含单个值；对于 ROW 关键字：一列，其中针对每一行包含一个值。	计算一系列单元格的最小值
STDV 或 STDEV	对于 ALL 关键字：新列，其中包含单个值；对于 COL 关键字：一列，其中针对每个输入列包含单个值；对于 ROW 关键字：一列，其中针对每一行包含一个值。	计算一系列单元格的标准差

## 数学函数和三角函数

宏名称	返回结果	描述
AVG	针对每个输入列返回一列	计算所指定数据范围内单元格的平均值或算术平均值
MAX	对于 ALL 关键字：新列，其中包含单个值；对于 COL 关键字：一列，其中针对每个输入列包含单个值；对于 ROW 关键字：一列，其中针对每一行包含一个值。	计算一系列单元格的最大值
MEAN	对于 ALL 关键字：新列，其中包含单个值；对于 COL 关键字：一列，其中针对每个输入列包含单个值；对于 ROW 关键字：一列，其中针对每一行包含一个值。	计算一系列单元格的平均值或算术平均值
MIN	对于 ALL 关键字：新列，其中包含单个值；对于 COL 关键字：一列，其中针对每个输入列包含单个值；对于 ROW 关键字：一列，其中针对每一行包含一个值。	计算一系列单元格的最小值
STDV 或 STDEV	对于 ALL 关键字：新列，其中包含单个值；对于 COL 关键字：一列，其中针对每个输入列包含单个值；对于 ROW 关键字：一列，其中针对每一行包含一个值。	计算一系列单元格的标准差

宏名称	返回结果	描述
SUM	对于 ALL 关键字：新列，其中包含单个值；对于 COL 关键字：一列，其中针对每个输入列包含单个值；对于 ROW 关键字：一列，其中针对每一行包含一个值。	计算某个范围的单元格的总和
TOTAL	对于 ALL 关键字：新列，其中包含单个值；对于 COL 关键字：一列，其中针对每个输入列包含单个值；对于 ROW 关键字：一列，其中针对每一行包含一个值。	计算某个范围的单元格的总和

## 字符串函数

宏名称	返回结果	描述
LIKE	针对每个输入列返回一列	确定文本字符串是否与指定的模式匹配
LOWER	针对每个输入列返回一列	将字符串值转换为小写
LTRIM	针对每个输入列返回一列	除去每个字符串值中的前导空格字符
NUMBER	针对每个输入列返回一列	将表示时间和日期的 ASCII 文本字符串转换为数字值
RTRIM	针对每个输入列返回一列	除去每个字符串值中的尾部空格字符
STRING_CONCAT	一个列，其中针对最短输入列中的每一行包含一个值	将所指定数据范围内的字符串并置
SUBSTR 或 SUBSTRING	针对每个输入列返回一列	返回字符串中从起始位置开始的字符
UPPER	针对每个输入列返回一列	将字符串值转换为大写

## 日期和时间函数

宏名称	返回结果	描述
CURRENT_DATE	针对每个输入列返回一列	以 format 格式返回当前日期
CURRENT_DAY	针对每个输入列返回一列	以 1-31 范围内的数字形式返回当天是一个月中的第几天
CURRENT_MONTH	针对每个输入列返回一列	以 1-12 范围内的数字形式返回一年内的当前月份
CURRENT_WEEKDAY	针对每个输入列返回一列	以 0-6 范围内的数字形式返回当天是一周中的第几天
CURRENT_YEAR	针对每个输入列返回一列	以数字形式返回当前年份
DATE	针对每个输入列返回一列	将日期字符串转换为儒略日期
DATE_FORMAT	针对每个输入列返回一列	转换日期格式

## 其他函数

宏名称	返回结果	描述
EXTERNALCALLOUT	由使用 ExternalCallout API 编写的定制应用程序定义的值	调用使用 ExternalCallout API 编写的定制应用程序。  有关更多信息，请参阅《IBM Interact 管理员指南》。
IF	一个列，其中针对最短输入列中的每一行包含一个值	开始条件式 if-then-else 语句

---

## 适用于 IBM Interact 的宏函数参数

本节描述 IBM Interact 中宏函数的参数和用法。

### 格式规范

本节描述一些常用参数的格式。该格式适用于本节中宏函数规范对这些参数进行的所有引用。

#### **data**

**data** 参数表示宏函数所要处理的数据列。此参数可以是常量或字段。请参阅特定的宏函数以了解详细信息。

注：IBM Interact 不支持同时对多个字段执行计算，也不支持对行的子集执行计算。

某些其他参数名称也使用与 **data** 相同的格式。这些参数的描述会引用此部分和格式。

#### **keyword**

**keyword** 参数控制宏函数的行为。它指出可以指定关键字（如果省略此参数，那么将使用缺省关键字）。在每个宏函数中，关键字选项以如下格式列出：

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

请选择提供所需行为的关键字选项。缺省选项以粗体显示。例如，给定下列选项：

```
{RADIANS | DEGREES}
```

下列宏函数均有效：

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

注：许多宏函数使用关键字参数 {ALL | COL | ROW}。这些关键字在 IBM Interact 中不适用，因为输入数据始终为单个列或字段。这个宏的行为始终像是已指定 COL 关键字一样。因此，使用 IBM Interact 时，无需指定这些关键字。

### 使用常量

大多数宏函数参数可使用数字常量或求值为数字常量的表达式（处理字符串的宏函数可使用字符串常量）。

在逐条记录地执行运算（例如，将两个数字列相加）的宏函数中，使用常量等同于指定一个其中每一行都包含该常量值的列。在本质上，提供常量作为输入参数时，该常量将扩展为长度与输入相同。

一些宏函数可使用 ASCII 文本字符串和数字常量。既可使用数字常量又可使用 ASCII 文本字符串的参数在每个宏函数的"参数"部分中作了说明。

下表中提供了一些示例。

函数定义	如何解释常量
PERCENT_UTILIZ = (CURR_BAL*100)/CREDIT_LIM	常量 100 解释为这样一个列：它包含与列 CURR_BAL 相同的行数，且每一行都包含常量 100。派生字段 PERCENT_UTILIZ 将包含 CURR_BAL 的每个值乘以 100 然后除以 CREDIT_LIM 的每个值所得到的结果。
NAME = STRING_CONCAT ("Mr.", LAST_NAME)	常量 "Mr." 解释为这样一个列：它包含与列 LAST_NAME 相同的行数，且每一行都包含常量 "Mr."。派生字段 NAME 将包含在 LAST_NAME 中的每个文本字符串前面添加 "Mr." 作为前缀后所得到的结果。

注：诸如 DT\_DELIM\_M\_D\_Y 之类的常量需要用单引号括起来。



## 第 3 章 IBM 宏参考

本节描述每个可用于 IBM Campaign 和/或 IBM Interact 的宏。宏按字母顺序列出。

**要点：**请勿将 IBM 宏语言中的函数名或关键字用于 IBM Campaign 中用户表的列标题，无论是从数据库还是平面文件进行映射均如此。在映射表的列标题中使用这些保留字将导致错误。

### 有效日期格式的关键字

下表显示有效格式的关键字以及每个关键字的描述和示例。

关键字	描述	示例
MM	2 位数月份	01、02、03、... 或 12
MMDD	2 位数月份和 2 位数日期	3 月 31 日表示为 0331
MMDDYY	2 位数月份、2 位数日期和 2 位数年份	1970 年 3 月 31 日表示为 033170
MMDDYYYY	2 位数月份、2 位数日期和 4 位数年份	1970 年 3 月 31 日表示为 03311970
DELIM_M_D	任何定界的月份和日子	March 31、3/31 或 03-31
DELIM_M_D_Y	任何定界的月份、日子和年份	March 31, 1970 或 3/31/70
DELIM_M_D_YYYY	任何定界的月份、日期和 4 位数年份	March 31, 1970 或 3/31/1970
DELIM_Y_M	任何定界的年份和月份	March, 70、3-70 或 3/1970
DELIM_Y_M_D	任何定界的年份、月份和日期	1970 Mar 31 或 70/3/31
YYMM	2 位数年份和三字母月份	70MAR
YYMMDD	2 位数年份、三字母月份和 2 位数日期	70MAR31
YY	2 位数年份	70
YYMM	2 位数年份和 2 位数月份	7003
YYMMDD	2 位数年份、2 位数月份和 2 位数日期	700331
YYYYMM	4 位数年份和三字母月份	1970MAR
YYYYMMDD	4 位数年份、三字母月份和 2 位数日期	1970MAR31
YYYY	4 位数的年份	1970
YYYYMM	4 位数年份和 2 位数月份	197003
YYYYMMDD	4 位数年份、2 位数月份和 2 位数日期	19700331
DELIM_M_Y	后跟年份的任何定界月份	3-70、3/70、Mar 70 或 March 1970
DELIM_D_M	后跟月份的任何定界日期	31-3、31/3 或 31 March
DELIM_D_M_Y	任何定界日期、月份和年份	31-MAR-70、31/3/1970 或 31 03 70
DD	2 位数日期	31

关键字	描述	示例
DDMMM	2 位数日期和三字母月份	31MAR
DDMMYY	2 位数日期、三字母月份和 2 位数年份	31MAR70
DDMMYYYY	2 位数日期、三字母月份和 4 位数年份	31MAR1970
DDMM	2 位数日期和 2 位数月份	3103
DDMMYY	2 位数日期、2 位数月份和 2 位数年份	310370
DDMMYYYY	2 位数日期、2 位数月份和 4 位数年份	31031970
MMYY	2 位数月份和 2 位数年份	0370
MMYYYY	2 位数月份和 4 位数年份	031970
MMM	三字母月份	三月
MMDD	三字母月份和 2 位数日期	MAR31
MMDDYY	三字母月份、2 位数日期和 2 位数年份	MAR3170
MMDDYYYY	三字母月份、2 位数日期和 4 位数年份	MAR311970
MMYY	3 字母月份和 2 位数年份	MAR70
MMYYYY	三字母月份和 4 位数年份	MAR1970
MONTH	一年中的月份	January、February、March 等等，或者 Jan、Feb、Mar 等等
WEEKDAY	星期几	Sunday、Monday、Tuesday 等等 (Sunday = 0)
WKD	星期几的缩写	Sun、Mon、Tues 等等 (Sun = 0)

## ABS 宏

ABS 宏仅在 IBM Campaign 中可用。

### 语法

ABS(data)

### 参数

data

要计算绝对值的数字值。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

ABS 计算所指定数据范围内数字的绝对值。数字的绝对值是其无符号值（即，正数保持不变，负数以正数返回）。ABS 针对每个输入列返回一个新列，其中每个新列都包含相应输入列中数字的绝对值。

## 示例

TEMP = ABS(-3) 或 TEMP = ABS(3)
创建名为 TEMP 的列，其中包含值 3。
TEMP = ABS(V1)
创建名为 TEMP 的列，其中每个值都是 V1 列内容的绝对值。
TEMP = ABS(V1:V3)
创建名称分别为 TEMP、VX 和 VY 的三个新列。TEMP 列中的值是 V1 列内容的绝对值，VX 列的值是 V2 列内容的绝对值，而 VY 列的值是 V3 列内容的绝对值。
TEMP = ABS(V1[10:20])
创建名为 TEMP 的列，其中，前 11 个单元格包含 V1 列第 10-20 行值的绝对值。TEMP 中的其他单元格为空。
TEMP = ABS(V1[1:5]:V2)
创建名称分别为 TEMP 和 VX 的两个新列，每个列在第 1-5 行都有值（其他单元格为空）。列 TEMP 中的值是列 V1 中相应行的绝对值，而列 VX 中的值是列 V2 中相应行的绝对值。

## 相关函数

函数	描述
SIGN	计算所指定数据范围内值的符号（正或负）。

## ACOS 宏

ACOS 宏仅在 IBM Campaign 中可用。

### 语法

ACOS( data [, units\_keyword ])

### 参数

data

要计算反余弦值的数字值。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

units\_keyword

这个可选关键字确定输入值和结果是解释为度数还是弧度数。请选择下列其中一个值：

RADIAN - 以弧度为单位执行计算（缺省情况）

DEGREE - 以度为单位执行计算

如果未指定此参数，那么缺省情况下将以弧度为单位。（要从弧度转换为度，请除以 PI 并乘以 180。）

有关在 IBM Campaign 中使用关键字的更多详细信息，请参阅第 6 页的『格式规范』。

## 描述

ACOS 计算所指定数据范围内值的反余弦值。反余弦是一个角度，其余弦值即为每个单元格的内容。ACOS 针对每个输入列返回一个新列，其中每个新列都包含相应输入列中数字的反余弦值。

如果使用关键字 RADIAN，那么 ACOS 将返回 0 到 $\pi$ 范围内的值。如果使用关键字 DEGREE，那么 ACOS 将返回 0 - 180 范围内的值。

注：每个指定列的单元格内容都必须介于 -1.0 与 1.0（含本数）之间的值。否则，对于每个无效输入，将返回空白单元格。

## 示例

TEMP = ACOS( $\theta$ )、TEMP = ACOS( $\theta$ , $\theta$ ) 或 TEMP = ACOS( $\theta$ , RADIAN)
创建名为 TEMP 的列，其中包含值 1.571 ( $\pi/2$ 弧度)。
TEMP = ACOS( $\theta$ , 1) 或 TEMP = ACOS( $\theta$ , DEGREE)
创建名为 TEMP 的列，其中包含值 90 (度)。
TEMP = ACOS(V1)
创建名为 TEMP 的列，其中每个值都是 V1 列内容的反余弦值（以弧度计）。
TEMP = ACOS(V1:V3, 1)
创建名称分别为 TEMP、VX 和 VY 的三个新列。TEMP 列中的值是 V1 列内容的反余弦值，VX 列的值是 V2 列内容的反余弦值，而 VY 列的值是 V3 列内容的反余弦值。所有值都以度为单位。
TEMP = ACOS(V1[10:20])
创建名为 TEMP 的新列，其中，前 11 个单元格包含 V1 列第 10-20 行值的反余弦值（以弧度计）。TEMP 中的其他单元格为空。
TEMP = ACOS(V1[1:5]:V2)
创建名称分别为 TEMP 和 VX 的两个新列，每个列在第 1-5 行都有值（其他单元格为空）。列 TEMP 中的值是列 V1 中相应行的反余弦值，而列 VX 中的值是列 V2 中相应行的反余弦值。所有值都以弧度为单位。

## 相关函数

函数	描述
ACOT	计算所指定数据范围中内容的反余切值
ASIN	计算所指定数据范围中内容的反正弦值
ATAN	计算所指定数据范围中内容的反正切值

函数	描述
COS	计算所指定数据范围中内容的余弦值

## ACOT 宏

ACOT 宏仅在 IBM Campaign 中可用。

### 语法

ACOT(data [, units\_keyword])

### 参数

data

要计算反余切值的数字值。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

units\_keyword

这个可选关键字确定输入值和结果是解释为度数还是弧度数。请选择下列其中一项：

RADIAN - 以弧度为单位执行计算（缺省情况）

DEGREE - 以度为单位执行计算

如果未指定此参数，那么缺省情况下将以弧度为单位。（要从弧度转换为度，请除以 PI 并乘以 180。）

有关在 IBM Campaign 中使用关键字的更多详细信息，请参阅第 6 页的『格式规范』。

### 描述

ACOT 返回一个角度，其余切值即为每个单元格的内容。ACOT 针对每个输入列返回一个新列，其中每个新列都包含相应输入列中数字的反余切值。使用 64 位浮点数。

### 示例

TEMP = ACOT(0.5)、TEMP = ACOT(0.5, 0) 或 TEMP = ACOT(0.5, RADIAN)
创建名为 TEMP 的列，其中包含值 2.157（弧度）。
TEMP = ACOT(1, 1) 或 TEMP = ACOT(1, DEGREE)
创建名为 TEMP 的列，其中包含值 0.022 (1/45) 度。
TEMP = ACOT(0)
创建名为 TEMP 的列，其中包含值 MAX32_Float（以弧度计）。
TEMP = ACOT(V1)
创建名为 TEMP 的新列，其中每个值都是 V1 列内容的反余切值（以弧度计）。

TEMP = ACOT(V1:V3, 1)
创建名称分别为 TEMP、VX 和 VY 的三个新列。TEMP 列中的值是 V1 列内容的反余切值，VX 列的值是 V2 列内容的反余切值，而 VY 列的值是 V3 列内容的反余切值。所有值都以度为单位。
TEMP = ACOT(V1[10:20])
创建名为 TEMP 的新列，其中，前 11 个单元格包含 V1 列第 10-20 行值的反余切值（以弧度计）。TEMP 中的其他单元格为空。
TEMP = ACOT(V1[1:5]:V2)
创建名称分别为 TEMP 和 VX 的两个新列，每个列在第 1-5 行都有值（其他单元格为空）。列 TEMP 中的值是列 V1 中相应行的反余切值，而列 VX 中的值是列 V2 中相应行的反余切值。所有值都以弧度为单位。

## 相关函数

函数	描述
ACOS	计算所指定数据范围中内容的反余弦值
ASIN	计算所指定数据范围中内容的反正弦值
ATAN	计算所指定数据范围中内容的反正切值
COT	计算所指定数据范围中内容的余切值

---

## ADD\_MONTHS 宏

ADD\_MONTHS 宏在 IBM Campaign 中可用。

### 语法

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

### 参数

months

一个整数，它表示要与 date\_string 相加的月份数。

date\_string

一个文本字符串，它表示有效日期，其格式为 DELIM\_M\_D\_Y，或者格式由可选的 input\_format 自变量指定。

input\_format

计算而得的日期所使用的格式。要获取受支持日期格式的列表，请参阅 DATE\_FORMAT 函数。请注意，input\_format 既确定输入字符串的格式，也确定输出字符串的格式。

### 描述

ADD\_MONTHS 返回将指定的月份数与指定的 date\_string 相加所得的日期。该日期将以缺省格式 (DELIM\_M\_D\_Y) 或者由可选 input\_format 自变量指定的格式返回。如果要使用另一种输出格式，请使用 DATE\_FORMAT。

如果将月份增加所指定月份数会生成无效日期，那么结果将计算为该月份的最后一天，如下面的最后一个示例所示。必要时，会将闰年考虑在内。例如，31-Jan-2012 加 1 个月将得到 29-Feb-2012。

## 示例

ADD\_MONTHS(12, '06-25-11') 将 1 年 (12 个月) 与指定的日期相加，并返回日期 06-25-12。

ADD\_MONTHS(3, '2011-06-25', DT\_DELIM\_Y\_M\_D) 将 3 个月与指定的日期相加，并返回日期 2011-09-25。

ADD\_MONTHS(1, '02-28-2011') 将返回日期 03-28-2011。

ADD\_MONTHS(1, '03-31-2012') 将返回日期 04-30-2012。

## 相关函数

函数	描述
DATE	将日期字符串转换为儒略日期。
DATE_FORMAT	将 <code>input_format</code> 格式的日期转换为 <code>output_format</code> 格式。

---

## AND 宏

AND 宏在 IBM Campaign 和 IBM Interact 中都可用。

### 语法

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

### 参数

data1

要与 data2 中的值进行逻辑 AND 运算的数字。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

data2

要与 data1 中的值进行逻辑 AND 运算的数字。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。除非 data2 是常量，否则 data2 中的列数必须与 data1 中的列数相等。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

### 描述

AND 计算两个所指定数据范围之间的逻辑 AND 运算结果。它针对每个输入列返回一个新列，其中每个新列都包含 data1 的相应列与 data2 的相应列进行逻辑 AND 运算的结果（即，data1 的第一列与 data2 的第一列进行逻辑 AND 运算，第二列与第二列进行逻辑 AND 运算，依此类推）。

如果 data2 是常量，那么 data1 中的每个值都与该值进行逻辑 AND 运算。如果 data2 包含一个或多个列，那么将在 data1 中的一列与 data2 中的一列之间逐行执行计算。data1 的第一行与 data2 的第一个行值进行逻辑 AND 运算，第二行与第二行进行逻辑 AND 运算，依此类推。这种逐行计算会针对每一行生成一个结果，直至处理完最短列中的最后一个值为止。

注：AND 运算符可以简写为 &&。使用 && 可以分隔两个自变量（例如，要指定 V1 AND 3，输入 V1&&3 即可）。

## 示例

TEMP = 1 AND 8 或 TEMP = 1 && 8
创建名为 TEMP 的新列，其中包含值 1（任何非零数字都视为 1）。
TEMP = V1 && 1
创建名为 TEMP 的新列，对于列 V1 的每个值，新列都包含值 1。
TEMP = V1 && V1
创建名为 TEMP 的新列，对于列 V1 的每个非零值，新列都包含值 1；对于列 V1 的每个零值，新列都包含值 0。
TEMP = V1 && V2
创建名为 TEMP 的新列，其中每个值都是列 V1 的行值与列 V2 的相应行值进行逻辑 AND 运算的结果。
TEMP = V1:V3 && V4:V6
创建名称分别为 TEMP、VX 和 VY 的三个新列。列 TEMP 包含 V1 中的值与列 V4 的相应行值进行逻辑 AND 运算的结果。列 VX 包含列 V2 与 V5 进行逻辑 AND 运算所生成的值。列 VY 包含列 V3 与 V6 进行逻辑 AND 运算所生成的值。
TEMP = V1[10:20] && V2 或 TEMP = V1[10:20] && V2[1:11]
创建名为 TEMP 的新列，其中，前 11 个单元格包含 V1 列第 10-20 行的值与 V2 列第 1-11 行的值进行逻辑 AND 运算的结果。TEMP 中的其他单元格为空。

## 相关函数

函数	描述
NOT	计算所指定数据范围中内容的逻辑 NOT 运算结果
OR	计算两个指定数据范围之间的逻辑 OR 运算结果

## ASIN 宏

ASIN 宏仅在 IBM Campaign 中可用。

### 语法

ASIN(data [, units\_keyword])

## 参数

data

要计算反正弦值的数字值。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

units\_keyword

这个可选关键字确定输入值和结果是解释为度数还是弧度数。请选择下列其中一项：

RADIAN - 以弧度为单位执行计算（缺省情况）

DEGREE - 以度为单位执行计算

如果未指定此参数，那么缺省情况下将以弧度为单位。（要从弧度转换为度，请除以 PI 并乘以 180。）

有关在 IBM Campaign 中使用关键字的更多详细信息，请参阅第 6 页的『格式规范』。

## 描述

ASIN 计算所指定数据范围内值的反正弦值。反正弦值是一个角度，其正弦值即为每个单元格的内容。ASIN 针对每个输入列返回一个新列，其中每个新列都包含相应输入列中数字的反正弦值。

如果使用关键字 RADIAN，那么 ASIN 将返回  $-\pi/2$  到  $\pi/2$  范围内的值。如果使用关键字 DEGREE，那么 ASIN 将返回 -90 到 90 范围内的值。

注：每个指定列的单元格内容都必须是介于 -1.0 与 1.0（含本数）之间的值。否则，对于每个无效输入，将返回 ???。

## 示例

TEMP = ASIN(0.5)、TEMP = ASIN(0.5, 0) 或 TEMP = ASIN(0.5, RADIAN)
创建名为 TEMP 的新列，其中包含值 0.524 ( $\pi/6$ 弧度)。
TEMP = ASIN(0.5, 1) 或 TEMP = ASIN(0.5, DEGREE)
创建名为 TEMP 的新列，其中包含值 30（度）。
TEMP = ASIN(V1)
创建名为 TEMP 的新列，其中每个值都是 V1 列内容的反正弦值（以弧度计）。
TEMP = ASIN(V1:V3, 1)
创建名称分别为 TEMP、VX 和 VY 的三个新列。TEMP 列中的值是 V1 列内容的反正弦值，VX 列的值是 V2 列内容的反正弦值，而 VY 列的值是 V3 列内容的反正弦值。所有值都以度为单位。
TEMP = ASIN(V1[10:20])
创建名为 TEMP 的新列，其中，前 11 个单元格包含 V1 列第 10-20 行值的反正弦值（以弧度计）。TEMP 中的其他单元格为空。

```
TEMP = ASIN(V1[1:5]:V2)
```

创建名称分别为 TEMP 和 VX 的两个新列，每个列在第 1-5 行都有值（其他单元格为空）。列 TEMP 中的值是列 V1 中相应行的反正弦值，而列 VX 中的值是列 V2 中相应行的反正弦值。所有值都以弧度为单位。

## 相关函数

函数	描述
ACOS	计算所指定数据范围中内容的反余弦值
ACOT	计算所指定数据范围中内容的反余切值
ATAN	计算所指定数据范围中内容的反正切值
SIN	计算所指定数据范围中内容的正弦值

---

## ATAN 宏

ATAN 宏仅在 IBM Campaign 中可用。

### 语法

```
ATAN(data [, units_keyword])
```

### 参数

data

要计算反正切值的数字值。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

units\_keyword

这个可选关键字确定输入值和结果是解释为度数还是弧度数。请选择下列其中一项：

RADIAN - 以弧度为单位执行计算（缺省情况）

DEGREE - 以度为单位执行计算

如果未指定此参数，那么缺省情况下将以弧度为单位。（要从弧度转换为度，请除以 PI 并乘以 180。）

有关在 IBM Campaign 中使用关键字的更多详细信息，请参阅第 6 页的『格式规范』。

### 描述

ATAN 计算所指定数据范围内值的反正切值。反正切值是一个角度，其正切值即为每个单元格的内容。ATAN 针对每个输入列返回一个新列，其中每个新列都包含相应输入列中数字的反正切值。

如果使用关键字 RADIAN，那么 ATAN 将返回  $-\pi/2$  到  $\pi/2$  范围内的值。如果使用关键字 DEGREE，那么 ATAN 将返回 -90 到 90 范围内的值。

## 示例

TEMP = ATAN(1)、TEMP = ATAN(1, 0) 或 TEMP = ATAN(1, RADIAN)
创建名为 TEMP 的新列，其中包含值 0.785 ( $\pi/4$ 弧度)。
TEMP = ATAN(1, 1) 或 TEMP = ATAN(1, DEGREE)
创建名为 TEMP 的新列，其中包含值 45 (度)。
TEMP = ATAN(V1)
创建名为 TEMP 的新列，其中每个值都是 V1 列内容的反正切值 (以弧度计)。
TEMP = ATAN(V1:V3, 1)
创建名称分别为 TEMP、VX 和 VY 的三个新列。TEMP 列中的值是 V1 列内容的反正切值，VX 列的值是 V2 列内容的反正切值，而 VY 列的值是 V3 列内容的反正切值。所有值都以度为单位。
TEMP = ATAN(V1[10:20])
创建名为 TEMP 的新列，其中，前 11 个单元格包含 V1 列第 10-20 行值的反正切值 (以弧度计)。TEMP 中的其他单元格为空。
TEMP = ATAN(V1[1:5]:V2)
创建名称分别为 TEMP 和 VX 的两个新列，每个列在第 1-5 行都有值 (其他单元格为空)。列 TEMP 中的值是列 V1 中相应行的反正切值，而列 VX 中的值是列 V2 中相应行的反正切值。所有值都以弧度为单位。

## 相关函数

函数	描述
ACOS	计算所指定数据范围中内容的反余弦值
ASIN	计算所指定数据范围中内容的反正弦值
ATAN	计算所指定数据范围中内容的反正切值
TAN	计算所指定数据范围中内容的正切值

---

## AVG 宏

AVG 宏在 IBM Campaign 和 IBM Interact 中都可用。

### 语法

AVG(data [, keyword])

### 参数

data

要计算算术平均值的数字值。这些值可以是常量值、列、单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

keyword

这个可选关键字确定如何对输入数据范围执行计算。请选择下列其中一个关键字：

ALL - 对 data 中的所有单元格执行计算（缺省值）

COL - 对 data 的每一列单独执行计算

ROW - 对 data 的每一行单独执行计算

有关在 IBM Campaign 中使用关键字的更多详细信息，请参阅第 6 页的『格式规范』。

注：许多宏函数使用关键字参数 {ALL | COL | ROW}。这些关键字在 **IBM Campaign** 中不适用，因为输入数据始终为单个列或字段。这个宏的行为始终像是已指定 COL 关键字一样。因此，使用 **IBM Campaign** 时，无需指定这些关键字。

## 描述

AVG 计算所指定数据范围内单元格的平均值或算术平均值。算术平均值的计算方法是，将所有单元格内容的总和除以单元格数。AVG 所返回的列数取决于 keyword。

- 如果 keyword 为 ALL，那么 AVG 将返回一个新列，其中包含单个值（data 中所有单元格的平均值）。
- 如果 keyword 为 COL，那么 AVG 将针对每个输入列返回一个新列。每个新列都包含一个值（相应输入列中所有单元格的平均值）。
- 如果 keyword 为 ROW，那么 AVG 将返回一个新列，其中包含 data 中每一行的平均值。

注：在计算时将会忽略空白单元格。

注：AVG 与 MEAN 宏函数等同。

## 示例

TEMP = AVG(V1)
创建名为 TEMP 的列，其中包含单个值，即 V1 列内容的算术平均值。
TEMP = AVG(V1:V3)
创建名为 TEMP 的列，其中包含单个值，即 V1、V2 和 V3 列内容的算术平均值。
TEMP = AVG(V1[10:20])
创建名为 TEMP 的列，其中包含单个值，即 V1 列第 10-20 行中各个单元格的算术平均值。
TEMP = AVG(V1[1:5]:V4)
创建名为 TEMP 的列，其中包含单个值，即 V1 到 V4 列第 1-5 行中各个单元格的算术平均值。

TEMP = AVG(V1:V3, COL)
创建名称分别为 TEMP、VX 和 VY 的三个新列。TEMP 列包含单个值，即 V1 列内容的算术平均值。VX 列包含单个值，即 V2 列内容的算术平均值。VY 列包含单个值，即 V3 列内容的算术平均值。
TEMP = AVG(V1[1:5]:V3, COL)
创建名称分别为 TEMP、VX 和 VY 的三个新列，其中每个列都包含单个值。TEMP 列中的值是 V1 列第 1-5 行中各个单元格的算术平均值。VX 列中的值是 V2 列第 1-5 行中各个单元格的算术平均值。VY 列中的值是 V3 列第 1-5 行中各个单元格的算术平均值。
TEMP = AVG(V1, ROW)
创建名为 TEMP 的列，其中包含与 V1 列相同的值（任何数字的算术平均值均为其自身）。
TEMP = AVG(V1:V3, ROW)
创建名为 TEMP 的列，其中每个单元格条目均为 V1、V2 和 V3 列中相应行的算术平均值。
TEMP = AVG(V1[1:5]:V3, ROW)
创建名为 TEMP 的列，其中 1-5 行的单元格包含 V1 到 V3 列中相应行的算术平均值。TEMP 中的其他单元格为空。

## 相关函数

函数	描述
SUM 或 TOTAL	计算某个范围的单元格的总和

## BETWEEN 宏

BETWEEN 宏仅在 IBM Campaign 中可用。

### 语法

value1 BETWEEN value2 AND value3

### 参数

等同于 value1 >= value2 AND < value3

### 描述

BETWEEN 是比较谓词的特殊变体。此谓词的详细信息非常重要，并且操作数的顺序有一些意外的影响。请参阅示例部分。

注：FROM 和 FOR 使用完全相同的语法。

### 示例

10 BETWEEN 5 AND 15 为 true，但是 10 BETWEEN 15 AND 5 为 false：

这是因为，使用 AND 来表达 BETWEEN 的这种等同方式具有特定的顺序，该顺序在您使用字面值时并不重要，但在您通过主变量、参数甚至子查询来提供 value2 和 value3 时至关重要。

## BIT\_AND 宏

BIT\_AND 宏在 IBM Campaign 和 IBM Interact 中都可用。

### 语法

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

### 参数

data1

要与 data2 中的值进行按位 AND 运算的非负整数。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

data2

要与 data1 中的值进行按位 AND 运算的非负整数。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。除非 data2 是常量，否则 data2 中的列数必须与 data1 中的列数相等。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

### 描述

BIT\_AND 在两个所指定数据范围之间执行按位 AND 运算。它针对每个输入列返回一个新列，其中每个新列都包含 data1 的相应列与 data2 的相应列进行按位 AND 运算的结果（即，data1 的第一列与 data2 的第一列进行按位 AND 运算，第二列与第二列进行按位 AND 运算，依此类推）。

如果 data2 是常量，那么 data1 中的每个值都与该值进行按位 AND 运算。如果 data2 包含一个或多个列，那么将在 data1 中的一列与 data2 中的一列之间逐行执行计算。data1 的第一行与 data2 的第一个行值进行按位 AND 运算，第二行与第二行进行按位 AND 运算，依此类推。这种逐行计算会针对每一行生成一个结果，直至处理完最短列中的最后一个值为止。

注：这个宏函数的精度限制为小于  $2^{24}$  的整数值。不允许使用负值。

注：BIT\_AND 运算符可以简写为 &。使用 & 可以分隔两个自变量（例如，要指定 BIT\_AND(V1, 3)，输入 V1&3 即可）。

### 示例

```
TEMP = 3 BIT_AND 7 或 TEMP = 3 & 7
```

创建名为 TEMP 的新列，其中包含值 3 (011 与 111 的按位 AND 运算结果为 011)。

```
TEMP = V1 & 8
```

创建名为 TEMP 的新列，其中每个值都是列 V1 的内容与二进制值 1000 进行按位 AND 运算的结果。

TEMP = V1 & V1
创建名为 TEMP 的新列，其中包含与 V1 列相同的内容（每个值与其自身执行 AND 运算会生成其自身）。
TEMP = V1 & V2
创建名为 TEMP 的新列，其中每个值都是列 V1 的行值与列 V2 的相应行值进行按位 AND 运算的结果。
TEMP = V1:V3 & V4:V6
创建名称分别为 TEMP、VX 和 VY 的三个新列。列 TEMP 包含 V1 中的值与列 V4 的相应行值进行按位 AND 运算的结果。列 VX 包含列 V2 与 V5 进行按位 AND 运算所生成的值。列 VY 包含列 V3 与 V6 进行按位 AND 运算所生成的值。
TEMP = V1[10:20] & V2 或 TEMP = V1[10:20] & V2[1:11]
创建名为 TEMP 的新列，其中，前 11 个单元格包含 V1 列第 10-20 行的值与 V2 列第 1-11 行的值进行按位 AND 运算的结果。TEMP 中的其他单元格为空。

## 相关函数

函数	描述
BIT_NOT	计算所指定数据范围中内容的按位 NOT 运算结果
BIT_OR	计算两个指定数据范围之间的按位 OR 运算结果
BIT_XOR 或 XOR	计算两个指定数据范围之间的按位 XOR 运算结果

## BIT\_NOT 宏

BIT\_NOT 宏在 IBM Campaign 和 IBM Interact 中都可用。

### 语法

BIT\_NOT data ~ data

### 参数

data

要执行按位 NOT 运算的非负整数。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

### 描述

BIT\_NOT 计算所指定数据范围内值的按位 NOT 运算结果。它针对每个输入列返回一个新列，其中每个新列都包含 data 的相应列中值的按位 NOT 运算结果。

注：这个宏函数的精度限制为小于  $2^{24}$  的整数值。不允许使用负值。

注：使用每一行中的数字  $x$  都与 data 相同的列等同于使用常量  $x$  作为 data。

注：BIT\_NOT 运算符可以简写为  $\sim$ 。请在数据值之前使用  $\sim$  符（例如，要指定 BIT\_NOT (V1)，输入  $\sim$ V1 即可）。

## 示例

TEMP = BIT_NOT 3 或 TEMP = $\sim$ 3
创建名为 TEMP 的新列，其中包含值 4 (011 的按位 NOT 运算结果为 100)。
TEMP = $\sim$ V1
创建名为 TEMP 的新列，其中每个值都是 V1 列内容的按位 NOT 运算结果。
TEMP = $\sim$ V1:V3
创建名称分别为 TEMP、VX 和 VY 的三个新列。TEMP 列中的值是 V1 列内容的按位 NOT 运算结果，VX 列的值是 V2 列内容的按位 NOT 运算结果，而 VY 列的值是 V3 列内容的按位 NOT 运算结果。
TEMP = $\sim$ V1[100:200]
创建名为 TEMP 的新列，其中，前 101 个单元格包含 V1 列第 1-50 行值的按位 NOT 运算结果。

## 相关函数

函数	描述
BIT_AND	计算两个指定数据范围之间的按位 AND 运算结果
BIT_OR	计算两个指定数据范围之间的按位 OR 运算结果
BIT_XOR 或 XOR	计算两个指定数据范围之间的按位 XOR 运算结果

---

## BIT\_OR 宏

BIT\_OR 宏在 IBM Campaign 和 IBM Interact 中都可用。

### 语法

data1 BIT\_OR data2 data1 或 data2 data1 | data2

### 参数

data1

要与 data2 中的值进行按位 OR 运算的非负整数。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

data2

要与 data1 中的值进行按位 OR 运算的非负整数。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。除非 data2 是常量，否则 data2 中的列数必须与 data1 中的列数相等。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

## 描述

BIT\_OR 在两个所指定数据范围之间执行按位 OR 运算。它针对每个输入列返回一个新列，其中每个新列都包含 data1 的相应列与 data2 的相应列进行按位 OR 运算的结果（即，data1 的第一列与 data2 的第一列进行按位 OR 运算，第二列与第二列进行按位 OR 运算，依此类推）。

如果 data2 是常量，那么 data1 中的每个值都与该值进行按位 OR 运算。如果 data2 包含一个或多个列，那么将在 data1 中的一列与 data2 中的一列之间逐行执行计算。data1 的第一行与 data2 的第一个行值进行按位 OR 运算，第二行与第二行进行按位 OR 运算，依此类推。这种逐行计算会针对每一行生成一个结果，直至处理完最短列中的最后一个值为止。

注：这个宏函数的精度限制为小于  $2^{24}$  的整数值。不允许使用负值。

注：BIT\_OR 运算符可以简写为竖线 (|)。使用竖线可以分隔两个列（例如，要指定 BIT\_OR(V1, 3)，输入 V1|3 即可）。另外，也可使用 OR。

## 示例

TEMP = 3 BIT_OR 7、TEMP = 3 OR 7 或 TEMP = 3   7
创建名为 TEMP 的新列，其中包含值 7（011 与 111 的按位 OR 运算结果为 111）。
TEMP = V1   8
创建名为 TEMP 的新列，其中每个值都是列 V1 的内容与二进制值 1000 进行按位 OR 运算的结果。
TEMP = V1   V1
创建名为 TEMP 的新列，其中包含与 V1 列相同的内容（每个值与其自身执行 OR 运算会生成其自身）。
TEMP = V1   V2
创建名为 TEMP 的新列，其中每个值都是列 V1 的行值与列 V2 的相应行值进行按位 OR 运算的结果。
TEMP = V1:V3   V4:V6
创建名称分别为 TEMP、VX 和 VY 的三个新列。列 TEMP 包含 V1 中的值与列 V4 的相应行值进行逻辑 OR 运算的结果。列 VX 包含列 V2 与 V5 进行逻辑 OR 运算所生成的值。列 VY 包含列 V3 与 V6 进行逻辑 OR 运算所生成的值。
TEMP = V1[10:20]   V2 或 TEMP = V1[10:20]   V2[1:11]
创建名为 TEMP 的新列，其中，前 11 个单元格包含 V1 列第 10-20 行的值与 V2 列第 1-11 行的值进行按位 OR 运算的结果。TEMP 中的其他单元格为空。

## 相关函数

函数	描述
BIT_AND	计算两个指定数据范围之间的按位 AND 运算结果
BIT_NOT	计算所指定数据范围中内容的按位 NOT 运算结果
BIT_XOR 或 XOR	计算两个指定数据范围之间的按位 XOR 运算结果

---

## BIT\_XOR 宏

BIT\_XOR 宏在 IBM Campaign 和 IBM Interact 中都可用。

### 语法

```
data1 BIT_XOR data2
```

### 参数

data1

要与 data2 中的值进行按位 XOR 运算的非负整数。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

data2

要与 data1 中的值进行按位 XOR 运算的非负整数。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。除非 data2 是常量，否则 data2 中的列数必须与 data1 中的列数相等。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

### 描述

BIT\_XOR 在两个所指定数据范围之间执行按位 XOR 运算。它针对每个输入列返回一个新列，其中每个新列都包含 data1 的相应列与 data2 的相应列进行按位 XOR 运算的结果（即，data1 的第一列与 data2 的第一列进行按位 XOR 运算，第二列与第二列进行按位 XOR 运算，依此类推）。

如果 data2 是常量，那么 data1 中的每个值都与该值进行按位 XOR 运算。如果 data2 包含一个或多个列，那么将在 data1 中的一列与 data2 中的一列之间逐行执行计算。data1 的第一行与 data2 的第一个行值进行按位 XOR 运算，第二行与第二行进行按位 XOR 运算，依此类推。这种逐行计算会针对每一行生成一个结果，直至处理完最短列中的最后一个值为止。

注：这个宏函数的精度限制为小于  $2^{24}$  的整数值。不允许使用负值。

## 示例

TEMP = 3 BIT_XOR 7
创建名为 TEMP 的新列，其中包含值 4 (011 与 111 的按位 XOR 运算结果为 100)。
TEMP = V1 BIT_XOR 8
创建名为 TEMP 的新列，其中每个值都是列 V1 的内容与二进制值 1000 进行按位 XOR 运算的结果。
TEMP = V1 BIT_XOR V1
创建名为 TEMP 的新列，其中包含的值全为零 (每个值与其自身执行 XOR 运算会生成零值)。
TEMP = V1 BIT_XOR V2
创建名为 TEMP 的新列，其中每个值都是列 V1 的行值与列 V2 的相应行值进行按位 XOR 运算的结果。
TEMP = V1:V3 BIT_XOR V4:V6
创建名称分别为 TEMP、VX 和 VY 的三个新列。列 TEMP 包含 V1 中的值与列 V4 的相应行值进行按位 XOR 运算的结果。列 VX 包含列 V2 与 V5 进行按位 XOR 运算所生成的值。列 VY 包含列 V3 与 V6 进行按位 XOR 运算所生成的值。
TEMP = V1[10:20] BIT_XOR V2 或 TEMP = V1[10:20] BIT_XOR V2[1:11]
创建名为 TEMP 的新列，其中，前 11 个单元格包含 V1 列第 10-20 行的值与 V2 列第 1-11 行的值进行按位 XOR 运算的结果。TEMP 中的其他单元格为空。

## 相关函数

函数	描述
BIT_AND	计算两个指定数据范围之间的按位 AND 运算结果
BIT_NOT	计算所指定数据范围中内容的按位 NOT 运算结果
BIT_OR	计算两个指定数据范围之间的按位 OR 运算结果

---

## CEILING 宏

CEILING 宏仅在 IBM Campaign 中可用。

### 语法

CEILING(data)

### 参数

data

要计算向上取整的数字值。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

## 描述

CEILING 计算所指定数据范围内值的向上取整。数字的向上取整是不小于该数的最小整数。CEILING 针对每个输入列返回一个新列，其中每个新列都包含相应输入列中数字的向上取整。

## 示例

TEMP = CEILING(4.3)
创建名为 TEMP 的新列，其中包含值 5。
TEMP = CEILING(-2.9)
创建名为 TEMP 的新列，其中包含值 -2。
TEMP = CEILING(V1)
创建名为 TEMP 的新列，其中每个值都是 V1 列内容的向上取整。
TEMP = CEILING(V1:V3)
创建名称分别为 TEMP、VX 和 VY 的三个新列。TEMP 列中的值是 V1 列内容的向上取整，VX 列的值是 V2 列内容的向上取整，而 VY 列的值是 V3 列内容的向上取整。
TEMP = CEILING(V1[10:20])
创建名为 TEMP 的新列，其中，前 11 个单元格包含 V1 列第 10-20 行值的向上取整。TEMP 中的其他单元格为空。
TEMP = CEILING(V1[50:99]:V2)
创建名称分别为 TEMP 和 VX 的两个新列，每个列在第 1-50 行都有值（其他单元格为空）。列 TEMP 中的值是列 V1 中各行的向上取整，而列 VX 中的值是列 V2 中各个值的向上取整。

## 相关函数

函数	描述
FLOOR 或 INT	计算所指定数据范围内每个值的向下取整
FRACTION	返回所指定数据范围内每个值的小数部分
TRUNCATE	返回所指定数据范围内每个值的非小数部分

---

## COLUMN 宏

COLUMN 宏仅在 IBM Campaign 中可用。

### 语法

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

### 参数

data

要用来创建列的值。此参数可以是常量值（数字或者括在引号内的 ASCII 文本）、列、单元格范围或求值为上述任何类型的表达式。此参数可以重复多次，但后续参数必须

具有与第一个参数相同的维度（即，列宽）。所有 data 参数中的所有值都必须是数字或 ASCII 文本（即，不可混合使用数字和文本值）。如果提供多个 data 参数，它们的列数必须相同。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

## 描述

COLUMN 将其输入垂直并置到函数组的列中。它返回数目与每个输入参数所提供列数相同的新列。您可提供无限数量的自变量。所有自变量都必须是数字或 ASCII 文本字符串（即，不可混合使用数字和文本值）。

注：COLUMN 宏函数可以简写，方法如下：在圆括号中列出以逗号分隔的 data 自变量，例如，TEMP = MEAN((1,2,3,4), ALL)。如果并非在另一个宏函数中使用，那么圆括号对并非必需（例如，V1=1,2,3 等同于 V1=COLUMN(1,2,3)）。

## 示例

TEMP = COLUMN(3, 4, 5)、TEMP = (3,4,5) 或 TEMP = 3,4,5
创建名为 TEMP 的新列，其中的前三个单元格包含值 3、4 和 5。
TEMP = COLUMN("one", "two", "three")
创建名为 TEMP 的新列，其中的前三个单元格包含值 "one"、"two" 和 "three"。
TEMP = AVG(V1), STDV(V1)
创建名为 TEMP 的新列，其中，第一个单元格包含 V1 列的平均值，第二个单元格包含 V1 列的标准差。
TEMP = V1:V2, V3:V4
创建名称分别为 TEMP 和 VX 的两个新列，其中，列 TEMP 包含列 V1 中的值，后跟列 V3 中的值。列 VX 包含列 V2 中的值，后跟列 V4 中的值。
TEMP = V1:V2, V3:V4
创建名称分别为 TEMP 和 VX 的两个新列，其中，列 TEMP 包含列 V1 中单元格 1-10 的值，后跟列 V3 中的所有值。列 VX 包含列 V2 中单元格 1-10 的值，后跟列 V4 中的所有值。
TEMP = V1:V2, V3:V4
创建名称分别为 TEMP 和 VX 的两个新列，每个列都包含单个值。列 TEMP 包含列 V1 和 V2 的平均值。列 VX 包含列 V3 和 V4 的平均值。

---

## COS 宏

COS 宏仅在 IBM Campaign 中可用。

### 语法

COS(data [, units\_keyword])

### 参数

data

要计算余弦值的数字值。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

units\_keyword

这个可选关键字确定输入值和结果是解释为度数还是弧度数。请选择下列其中一项：

RADIAN - 以弧度为单位执行计算（缺省情况）

DEGREE - 以度为单位执行计算

如果未指定此参数，那么缺省情况下将以弧度为单位。（要从弧度转换为度，请除以 PI 并乘以 180。）

有关在 IBM Campaign 中使用关键字的更多详细信息，请参阅第 6 页的『格式规范』。

## 描述

COS 计算所指定数据范围内值的余弦值。它针对每个输入列返回一个新列，其中每个新列都包含相应输入列中数字的余弦值。

## 示例

TEMP = COS(PI)、TEMP = COS(PI, 0) 或 TEMP = COS(PI, RADIAN)
返回名为 TEMP 的新列，其中包含单个值 -1。
TEMP = COS(90, 1) 或 TEMP = COS(90, DEGREE)
返回名为 TEMP 的新列，其中包含单个值 0。
TEMP = COS(V1)、TEMP = COS(V1, 0) 或 TEMP = COS(V1, RADIAN)
创建名为 TEMP 的新列，其中每个值都是 V1 列内容的余弦值（以弧度计）。
TEMP = COS(V1:V3, 1)
创建名称分别为 TEMP、VX 和 VY 的三个新列。TEMP 列中的值是 V1 列内容的余弦值，VX 列的值是 V2 列内容的余弦值，而 VY 列的值是 V3 列内容的余弦值。所有值都以度为单位。
TEMP = COS(V1[10:20])
创建名为 TEMP 的新列，其中，前 11 个单元格包含 V1 列第 10-20 行值的余弦值（以弧度计）。TEMP 中的其他单元格为空。
TEMP = COS(V1[1:5]:V2)
创建名称分别为 TEMP 和 VX 的两个新列，每个列在第 1-5 行都有值（其他单元格为空）。列 TEMP 中的值是列 V1 中相应行的余弦值，而列 VX 中的值是列 V2 中相应行的余弦值。所有值都以弧度为单位。

## 相关函数

函数	描述
ACOS	计算所指定数据范围中内容的反余弦值

函数	描述
COSH	计算所指定数据范围中内容的双曲余弦值
SIN	计算所指定数据范围中内容的正弦值
TAN	计算所指定数据范围中内容的正切值

## COSH 宏

COSH 宏仅在 IBM Campaign 中可用。

### 语法

COSH(data [, units\_keyword])

### 参数

data

要计算双曲余弦值的数字值。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

units\_keyword

这个可选关键字确定输入值和结果是解释为度数还是弧度数。请选择下列其中一项：

RADIAN - 以弧度为单位执行计算（缺省情况）

DEGREE - 以度为单位执行计算

如果未指定此参数，那么缺省情况下将以弧度为单位。（要从弧度转换为度，请除以 PI 并乘以 180。）

有关在 IBM Campaign 中使用关键字的更多详细信息，请参阅第 6 页的『格式规范』。

### 描述

COSH 计算所指定数据范围内值的双曲余弦值。对于以弧度为单位的  $x$ ，数字的双曲余弦值为：

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

其中  $e$  是自然数 2.7182818。COSH 针对每个输入列返回一个新列，其中每个新列都包含相应输入列中数字的双曲余弦值。

注：当值  $x$  过大时，将会返回溢出错误。 $\cosh(x)$  超过最大 32 位浮点值时就会发生这种情况。

## 示例

TEMP = COSH(0)、TEMP = COSH(0, 0) 或 TEMP = COSH(0, RADIAN)
返回名为 TEMP 的新列，其中包含值 1。
TEMP = COSH(V1)
创建名为 TEMP 的新列，其中每个值都是 V1 列内容的双曲余弦值（以弧度计）。
TEMP = COSH(V1:V3, 1) 或 TEMP = COSH(V1:V3, DEGREE)
创建名称分别为 TEMP、VX 和 VY 的三个新列。TEMP 列中的值是 V1 列内容的双曲余弦值，VX 列的值是 V2 列内容的双曲余弦值，而 VY 列的值是 V3 列内容的双曲余弦值。所有值都以度为单位。
TEMP = COSH(V1[10:20])
创建名为 TEMP 的新列，其中，前 11 个单元格包含 V1 列第 10-20 行值的双曲余弦值（以弧度计）。TEMP 中的其他单元格为空。
TEMP = COSH(V1[1:5]:V2)
创建名称分别为 TEMP 和 VX 的两个新列，每个列在第 1-5 行都有值（其他单元格为空）。列 TEMP 中的值是列 V1 中相应行的双曲余弦值，而列 VX 中的值是列 V2 中相应行的双曲余弦值。所有值都以弧度为单位。

## 相关函数

函数	描述
ACOS	计算所指定数据范围中内容的反余弦值
COS	计算所指定数据范围中内容的余弦值
SINH	计算所指定数据范围中内容的双曲正弦值
TANH	计算所指定数据范围中内容的双曲正切值

---

## COT 宏

COT 宏仅在 IBM Campaign 中可用。

### 语法

COT(data [, units\_keyword])

### 参数

data

要计算余切值的数字值。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

units\_keyword

这个可选关键字确定输入值和结果是解释为度数还是弧度数。请选择下列其中一项：

RADIAN - 以弧度为单位执行计算（缺省情况）

DEGREE - 以度为单位执行计算

如果未指定此参数，那么缺省情况下将以弧度为单位。（要从弧度转换为度，请除以 PI 并乘以 180。）

有关在 IBM Campaign 中使用关键字的更多详细信息，请参阅第 6 页的『格式规范』。

## 描述

COT 计算所指定数据范围内值的余切值。余切值是正切值的倒数。COT 针对每个输入列返回一个新列，其中每个新列都包含相应输入列中数字的余切值。

注：如果单元格包含正切值为零的值，那么反余切值为无穷大。在这种情况下，COT 将返回最大的 32 位浮点数。

## 示例

TEMP = COT(90)、TEMP = COT(90, 0) 或 TEMP = COT(90, RADIAN)
返回名为 TEMP 的新列，其中包含值 -0.5。
TEMP = COT(0)
返回名为 TEMP 的新列，其中包含值 MAX_FLOAT_32。
TEMP = COT(V1, 1) 或 TEMP = COT(V1, DEGREE)
创建名为 TEMP 的新列，其中每个值都是 V1 列内容的余切值（以度计）。
TEMP = COT(V1:V3, 1)
创建名称分别为 TEMP、VX 和 VY 的三个新列。TEMP 列中的值是 V1 列内容的余切值，VX 列的值是 V2 列内容的余切值，而 VY 列的值是 V3 列内容的余切值。所有值都以度为单位。
TEMP = COT(V1[10:20])
创建名为 TEMP 的新列，其中，前 11 个单元格包含 V1 列第 10-20 行值的余切值（以弧度计）。TEMP 中的其他单元格为空。
TEMP = COT(V1[1:5]:V2)
创建名称分别为 TEMP 和 VX 的两个新列，每个列在第 1-5 行都有值（其他单元格为空）。列 TEMP 中的值是列 V1 中相应行的余切值，而列 VX 中的值是列 V2 中相应行的余切值。所有值都以弧度为单位。

## 相关函数

函数	描述
ACOT	计算所指定数据范围中内容的反余切值
COS	计算所指定数据范围中内容的余弦值
SIN	计算所指定数据范围中内容的正弦值
TAN	计算所指定数据范围中内容的正切值

## COUNT 宏

COUNT 宏仅在 IBM Campaign 中可用。

### 语法

COUNT(data)

### 参数

data

要计算单元格数量的单元格范围。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

### 描述

COUNT 计算所指定数据范围内值的数量。它返回一个新列，其中包含单个值，表示所指定数据范围内包含值的单元格数量。

注：对空白列进行计数将返回零。

### 示例

TEMP = COUNT(AVG(V1:V5))
创建名为 TEMP 的新列，其中包含单个值 1（在缺省方式下，函数 AVG 返回单个单元格）。
TEMP = COUNT(V1)
创建名为 TEMP 的新列，其中包含单个值，用于指出 V1 列中包含值的单元格数量。
TEMP = COUNT(V1:V3)
创建名为 TEMP 的新列，其中包含单个值，指出 V1、V2 和 V3 列中包含值的单元格数量。
TEMP = COUNT(V1[10:20])
创建名为 TEMP 的新列，其中包含值 11（范围含括上下限值），前提是所有单元格都包含值。
TEMP = COUNT(V1[1:5]:V4)
创建名为 TEMP 的新列，其中包含值 20（每一列有 5 个单元格 x 4 列 = 20 个单元格），前提是所有单元格都包含值。
TEMP = COUNT(V1[1:10])
创建名为 TEMP 的新列，其中包含值 3，前提是 V1 列中第 1-3 行包含值，第 4-10 行为空。

### 相关函数

函数	描述
SUM 或 TOTAL	计算某个范围的单元格的总和

## CURRENT\_DATE 宏

CURRENT\_DATE 宏在 IBM Campaign 和 IBM Interact 中都可用。

### 语法

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

### 参数

format

下表中的某个关键字，用于指定 date\_string 的日期格式。

注：有关可用日期格式的更多信息，请参阅『有效的格式关键字』。

### 描述

CURRENT\_DATE 以 format 格式返回当前日期。日期是由 IBM 服务器上的时钟确定。未提供 format 关键字时，将会使用缺省值 DELIM\_M\_D\_Y。

针对所有建议数据库，IBM Campaign 会尝试使用数据库支持的当前时间 SQL 调用（例如，SYSDATE、GETDATE、DATE 或 TODAY）在数据库中运行 CURRENT\_DATE 宏。在这些情况下，这个宏函数的所有参数（包括日期格式）都会被忽略，并且输出中包括数据库所返回的所有内容（例如，输出中可能包括时间部分）。如果发生这种情况，并且您希望仅返回日期或以另一格式返回日期，您可使用原始 SQL 编写自己的定制宏，或者使用其他 IBM 宏。例如：

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

在某些情况下，CURRENT\_DATE() 宏会在 IBM Campaign 服务器上运行（例如，针对平面文件运行，针对没有等效 SQL 支持的非建议数据库运行，或者在数据库中无法解析 Campaign 宏表达式）。在这些情况下可以识别所有参数，并且输出将以选定格式返回。

注：并非 IBM Campaign 中的所有可用格式都受 IBM Interact 支持。

请注意，您可能必须使用 DATE\_FORMAT 宏针对您的数据库类型来调整 CURRENT\_DATE。例如，以下宏可以与 DB2 配合使用：

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

但是，对于 Oracle，必须使用 DATE\_FORMAT 宏，如下所示：

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

### 示例

如果今天的日期是 2015 年 9 月 13 日，那么 CURRENT\_DATE() 将返回 "09/13/15"。

### 相关函数

函数	描述
DATE_FORMAT	将日期从一种格式转换为另一种格式。
DATE_JULIAN	返回输入的儒略日期。

函数	描述
DATE_STRING	返回儒略日期的日期字符串。
DATE	将日期字符串转换为儒略日期。

---

## CURRENT\_DAY 宏

CURRENT\_DAY 宏在 IBM Campaign 和 IBM Interact 中都可用。

### 语法

CURRENT\_DAY()

### 描述

CURRENT\_DAY 以 1-31 之间的数字形式返回当天是一个月中的第几天。日期是由 IBM 服务器上的系统时钟确定。

### 示例

如果今天的日期是 6 月 19 日，那么 CURRENT\_DAY() 将返回数字 19。

### 相关函数

函数	描述
CURRENT_JULIAN	返回当前日期的儒略数。
CURRENT_MONTH	以数字形式返回当前月份。
CURRENT_TIME	以字符串形式返回当前时间。
CURRENT_WEEKDAY	以数字形式返回当天是一周中的第几天。
CURRENT_YEAR	以数字形式返回当前年份。

---

## CURRENT\_JULIAN 宏

CURRENT\_JULIAN 宏仅在 IBM Campaign 中可用。

### 语法

CURRENT\_JULIAN()

### 描述

CURRENT\_JULIAN() 返回当前日期的儒略数（自 0000 年 1 月 1 日以来经过的天数）。这相当于 DATE(CURRENT\_DATE()) 宏。

### 示例

如果今天的日期是 2000 年 8 月 31 日，那么 CURRENT\_JULIAN() 将返回数字 730729。

## 相关函数

函数	描述
CURRENT_DAY	以数字形式返回当前日子。
CURRENT_MONTH	以数字形式返回当前月份。
CURRENT_TIME	以字符串形式返回当前时间。
CURRENT_WEEKDAY	以数字形式返回当天是一周中的第几天。
CURRENT_YEAR	以数字形式返回当前年份。

---

## CURRENT\_MONTH 宏

CURRENT\_MONTH 宏在 IBM Campaign 和 IBM Interact 中都可用。

### 语法

CURRENT\_MONTH()

### 描述

CURRENT\_MONTH 以 1-12 的数字形式返回一年内的当前月份。

### 示例

如果今天的日期是 6 月 19 日，那么 CURRENT\_MONTH() 将返回数字 6。

## 相关函数

函数	描述
CURRENT_DAY	以数字形式返回当前日子。
CURRENT_JULIAN	返回当前儒略数。
CURRENT_TIME	以字符串形式返回当前时间。
CURRENT_WEEKDAY	以数字形式返回当天是一周中的第几天。
CURRENT_YEAR	以数字形式返回当前年份。

---

## CURRENT\_TIME 宏

CURRENT\_TIME 宏仅在 IBM Campaign 中可用。

### 语法

CURRENT\_TIME()

### 描述

CURRENT\_TIME 以字符串形式返回当前时间。时间是由 IBM 服务器上的系统时钟确定。

## Web 应用程序的日期设置

要在当前版本 IBM Campaign 中的 Web 应用程序上正确显示日期，必须先正确配置后端服务器的配置文件。这对于包含系统表的数据库的 dDateFormat 和

DateOutputFormatString 参数尤其重要。如果未正确配置这些参数，那么日期在 Campaign 中也会无法正确显示。您可以使用 IBM Marketing Platform 来配置这些属性。

## 在 Web 应用程序中针对特定语言来设置日期

注：除非特别注明，否则引用的所有文件均由 Web 应用程序安装程序安装。

要点：webappphome 是指 Campaign Web 应用程序的安装目录。language\_code 是指您为系统选择的语言设置。

1. 编辑 webappphome/conf/campaign\_config.xml 文件，确保以逗号分隔列表的形式，在 <supportedLocales> 标记中指定 language\_code，如下所示：

```
<supportedLocales>en_US, language_code</supportedLocales>
```

2. 在 webappphome/webapp 目录中，将整个目录树 en\_US 复制到 language\_code（区分大小写）。
3. 在 webappphome/webapp/WEB-INF/classes/resources 中，将 StaticMessages\_en\_US.properties 复制到 StaticMessages\_language\_code.properties。此外，还要将 ErrorMessage\_en\_US.properties 复制到 ErrorMessage\_language\_code.properties。
4. 编辑 StaticMessages\_language\_code.properties：搜索 DatePattern 并将其更改为 DatePattern=dd/MM/yyyy（区分大小写）。

注：此格式是由 Java™ 定义。您可在 java.text.SimpleDateFormat 的相关 Java 文档中找到有关此格式的完整详细信息 (<http://java.sun.com>)。不需要修改 StaticMessages.properties 文件。

5. 对于 WebSphere®：重新打包 Web 应用程序。
6. 对于 WebLogic：除去当前 Web 应用程序模块。
  - a. 添加新模块。
  - b. 重新部署 Web 应用程序。
  - c. 无需重新启动 Campaign 侦听器。
7. 确保在 Web 浏览器的语言设置中，language\_code 已设置为第一优先级。有关更多详细信息，请参阅下列各节：『设置 Web 浏览器以使用正确的语言』和『设置计算机以显示特定语言』。

注：在 language\_code 中，请确保使用连字符代替下划线。只有在 Web 应用程序配置中，才使用连字符代替下划线。

8. 登录 Campaign。在 Campaign 中，日期应该会以 StaticMessages\_language\_code.properties 所指定的格式显示。

有关如何为 IBM Campaign 配置时间的信息，请参阅 *IBM Campaign* 文档。

### 示例

如果时间是上午 10:54，那么 CURRENT\_TIME() 将返回字符串"10:54:00 AM"。

## 相关函数

函数	描述
CURRENT_DAY	以数字形式返回当前日子。
CURRENT_JULIAN	返回当前儒略数。
CURRENT_WEEKDAY	以数字形式返回当天是一周中的第几天。
CURRENT_YEAR	以数字形式返回当前年份。

---

## CURRENT\_WEEKDAY 宏

CURRENT\_WEEKDAY 宏在 IBM Campaign 和 IBM Interact 中都可用。

### 语法

CURRENT\_WEEKDAY()

### 描述

CURRENT\_WEEKDAY 以 0-6 的数字形式返回当天是一周中的第几天。星期日表示为 0，星期一表示为 1，依此类推。

### 示例

如果今天是星期五，那么 CURRENT\_WEEKDAY() 将返回数字 5。

## 相关函数

函数	描述
CURRENT_DAY	以数字形式返回当前日子。
CURRENT_JULIAN	返回当前儒略数。
CURRENT_MONTH	以数字形式返回当前月份。
CURRENT_TIME	以字符串形式返回当前时间。
CURRENT_YEAR	以数字形式返回当前年份。

---

## CURRENT\_YEAR 宏

CURRENT\_YEAR 宏在 IBM Campaign 和 IBM Interact 中都可用。

### 语法

CURRENT\_YEAR()

### 描述

CURRENT\_YEAR 以数字形式返回当前年份。

### 示例

如果当前年份为 2000，那么 CURRENT\_YEAR() 将返回数字 2000。

## 相关函数

函数	描述
CURRENT_DAY	以数字形式返回当前日子。
CURRENT_JULIAN	返回当前儒略数。
CURRENT_MONTH	以数字形式返回当前月份。
CURRENT_TIME	以字符串形式返回当前时间。
CURRENT_WEEKDAY	以数字形式返回当天是一周中的第几天。
MONTHOF	以数字形式返回一年中的月份。
WEEKDAYOF	以数字形式返回某个日期是一周中的第几天。
YEAROF	以数字形式返回年份。

---

## DATE

### 语法

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

### 参数

date\_string

表示有效日期的文本字符串。

format

可选，『有效的日期格式关键字』的表中的某个关键字，用于指定 date\_string 的日期格式。

### 描述

Interact DATE 宏将输入日期转换为与格式无关的整数。

DATE 宏的计算方法如下： $DATE(X) = 365 +$  从 0001 年 1 月 1 日中午起经过的完整天数。对于 DATE (X)，可以提供可选的输入 DATE 格式关键字来指定如何解析输入 DATE。未提供 format 关键字时，将会使用缺省值 DELIM\_M\_D\_Y。有关更多信息，请参阅有效的日期格式关键字，以了解有效日期格式的其他相关信息。

日期格式为固定宽度（例如，1970 年 2 月 28 日以 MMDDYYYY 格式表示为 02281970），或者是定界格式（例如，以 DELIM\_M\_D\_YY 格式表示为 February 28, 1970、2-28-1970 或 02/28/1970）。

在定界格式中，分隔符为斜杠 (/)、短划线 (-)、空格 ( )、逗号 (,) 或冒号 (:); 年份可以表示为 2 位数或 4 位数；月份可以是完整拼写（例如 February）、缩写（例如，Feb）或数字（例如，2 或 02）。

对于指定为 2 位数的所有年份：

- 缺省情况下，Interact 假定定界 2 位数日期在 1920 年与 2020 年之间
- 小于千禧年分界值（缺省值为 20，但可以通过 JVM 参数来设置）的 2 位数年份视为二十一世纪年份。

- 大于或等于阈值的 2 位数年份视为二十世纪年份。

注：

- 并非 Campaign 中的所有可用日期格式都受 Interact 支持。
- 有关 2 位数年份的更多信息，请参阅有效的日期格式关键字。
- 有关配置 2 位数年份的千禧年分界值的更多信息，请参阅《Interact 调整指南》中的『JVM 自变量』一节。

这个宏在 IBM Interact 中可用。

许多业务系统使用儒略日期偏移。Interact DATE() 宏的结果与儒略日期相关，如下所示：

儒略日期 = DATE(...) + 1,721,059 + 前一天中午以来经过的小数天数。

A.D 时间段的有用 DATE() 值包括：

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

注：根据 ISO 8601 标准以及 Date 和 DateTime 对象的 XML 模式定义，使用前公历来计算经过的天数。在这个日历系统中，前公历年份 0000 A.D. 与 0001 B.C. 同义。

## 示例

DATE("8/31/2000") 返回数字 730,729。

DATE("8/31/2000",DELIM\_MM\_DD\_YYYY) 返回数字 730,729。

DATE("2015-01-01",DELIM\_Y\_M\_D) 返回数字 735,965。

DATE("01",DD), DATE("0101",MMDD) 和 DATE("1970-01-01",DELIM\_Y\_M\_D) 返回数字 719,529。

## 相关函数

函数	描述
DATE_FORMAT	将日期从一种格式转换为另一种格式。
DATE_JULIAN	返回输入的儒略日期。
DATE_STRING	返回儒略日期的日期字符串。
CURRENT_DATE	以指定格式返回当前日期。

---

## DATE\_FORMAT 宏

DATE\_FORMAT 宏在 IBM Campaign 和 IBM Interact 中都可用。

## 语法

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

## 参数

date\_string

表示有效日期的文本。

input\_format

下表中的某个关键字，用于指定 date\_string 的日期格式。

output\_format

下表中的某个关键字，用于指定所需的输出日期格式。

## 描述

DATE\_FORMAT() 将 input\_format 格式的日期转换为 output\_format 格式。

如果日期为固定宽度，此参数必须设置为下列其中一个值：

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

MM 是 2 位数的月份，MMM 是 3 个字符的月份缩写。

如果日期为定界格式（可以使用任何定界符，包括空格、短划线和斜杠），那么此参数必须设置为下列其中一个值：

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

注：并非 IBM Campaign 中的所有可用格式都受 IBM Interact 支持。

## 示例

DATE\_FORMAT("012171", MMDDYY, MMDDYYYY) 返回字符串"01211971"。

注：请参阅第 46 页的『DATE』，以获取有效日期格式的更多相关信息。

## 相关函数

函数	描述
DATE	将日期字符串转换为儒略日期。
DATE_JULIAN	返回输入的儒略日期。

函数	描述
DATE_STRING	返回儒略日期的日期字符串。

---

## DATE\_JULIAN 宏

DATE\_JULIAN 宏仅在 IBM Campaign 中可用。

### 语法

DATE\_JULIAN(year, month, day)

### 参数

year

有效的 2 位数或 4 位数年份编号。

month

介于 1-12 之间的有效月份编号。

day

介于 1-31 之间的有效日子编号。

### 描述

DATE\_JULIAN 返回所指定输入的儒略日期。儒略日期是 0000 年 1 月 1 日以来经过的天数。

### 示例

DATE\_JULIAN (2000,08,31) 返回数字 730729。

### 相关函数

函数	描述
DATE	将日期字符串转换为儒略日期。
DATE_FORMAT	将日期从一种格式转换为另一种格式。
DATE_STRING	返回儒略日期的日期字符串。

---

## DATE\_STRING 宏

DATE\_STRING 宏仅在 IBM Campaign 中可用。

### 语法

DATE\_STRING(julian\_date [, 'output\_format'[, max\_length]]) DATE\_STRING(julian\_date [, 'format\_string'[, max\_length]])

## 参数

julian\_date

一个数字，它表示儒略日期，即 0000 年 1 月 1 日以来经过的天数。

output\_format

一个字符串，它是有效日期格式。

max\_length

format\_string

一个格式字符串，它可选择性地包括下列格式代码的任意组合：

代码	描述
%a	缩写的工作日名称
%A	完整的工作日名称
%b	缩写的月份名称
%B	完整的月份名称
%c	适用于语言环境的日期及时间表示法
%d	一个月中的第几天 (01 - 31)
%H	24 小时格式的小时 (00 - 23)
%I	12 小时格式的小时 (01 - 12)
%j	一年中的第几天 (001 - 366)
%m	月份 (01 - 12)
%M	分钟 (00 - 59)
%p	当前语言环境下 12 小时制的 AM/PM 指示符
%S	秒 (00 - 59)
%U	一年中的第几周，星期日作为每周的第一天 (00 - 51)
%w	工作日 (0 - 6; 星期日为 0)
%W	一年中的第几周，星期一作为每周的第一天 (00 - 51)
%x	当前语言环境的日期表示
%X	当前语言环境的时间表示
%y	2 位数年份 (00 - 99)
%Y	4 位数年份。不会截断年份中的前置零。例如，年份 0201 显示为 0201，年份 0001 显示为 0001。
%4Y	4 位数年份。不会截断年份中的前置零。例如，年份 0201 显示为 0201，年份 0001 显示为 0001。
%z 和 %Z	时区名称或简写；如果时区未知，那么就没有输出
%%	百分比符号

## 描述

DATE\_STRING 返回儒略日期的日期字符串。如果未提供 output\_format，那么将使用缺省关键字 DELIM\_M\_D\_Y。

## 示例

DATE\_STRING(730729) 返回字符串 "08/31/00"。

注：请参阅 第 46 页的『DATE』，以获取有效日期格式的更多相关信息。

## 相关函数

函数	描述
DATE	将日期字符串转换为儒略日期。
DATE_JULIAN	返回输入的儒略日期。
DATE_FORMAT	将日期从一种格式转换为另一种格式。

---

## DAY\_BETWEEN 宏

DAY\_BETWEEN 宏仅在 IBM Campaign 中可用。

### 语法

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

### 参数

from\_date\_string

表示有效日期的文本，计算经过天数起始于该日期。

to\_date\_string

表示有效日期的文本，计算经过天数截止于该日期。此日期的格式必须与 from\_date\_string 相同。

input\_format

下表中的某个关键字，用于指定 from\_date\_string 和 to\_date\_string 的日期格式。

### 描述

DAY\_BETWEEN 返回 from\_date\_string 与 to\_date\_string 之间的天数。如果未提供 input\_format，那么将使用缺省关键字 DELIM\_M\_D\_Y。

### 示例

DAY\_BETWEEN("08/25/00","08/31/00") 返回数字 6。

注：请参阅 第 46 页的『DATE』，以获取有效日期格式的更多相关信息。

## 相关函数

函数	描述
DAY_FROMNOW	返回当前日期与指定日期之间的天数。
DAY_INTERVAL	返回两个指定日期之间的天数。

---

## DAY\_FROMNOW 宏

DAY\_FROMNOW 宏仅在 IBM Campaign 中可用。

### 语法

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

### 参数

to\_year

有效的 2 位数或 4 位数年份编号。

to\_month

介于 1-12 之间的有效月份编号。

to\_day

介于 1-31 之间的有效日子编号。

### 描述

DAY\_FROMNOW 返回当前日期与 to\_year/to\_month/to\_day 所指定日期之间的天数。

注：如果指定的日期是在过去，那么返回的值将是负数。

### 示例

如果今天的日期是 2000 年 8 月 31 日，那么 DAY\_FROMNOW(2000,12,31) 将返回数字 122。

## 相关函数

函数	描述
DAY_BETWEEN	返回两个指定日期字符串之间的天数。
DAY_INTERVAL	返回两个指定日期之间的天数。

---

## DAY\_INTERVAL 宏

DAY\_INTERVAL 宏仅在 IBM Campaign 中可用。

### 语法

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

## 参数

from\_year

有效的 2 位数或 4 位数年份编号。

from\_month

介于 1-12 之间的有效月份编号。

from\_day

介于 1-31 之间的有效日子编号。

to\_year

有效的 2 位数或 4 位数年份编号。

to\_month

介于 1-12 之间的有效月份编号。

to\_day

介于 1-31 之间的有效日子编号。

## 描述

DAY\_INTERVAL 返回指定开始日期 (from\_year/from\_month/from\_day) 与指定结束日期 (to\_year/to\_month/to\_day) 之间的天数。

## 示例

DAY\_INTERVAL(2000,8,31,2000,12,31) 返回数字 122。

## 相关函数

函数	描述
DAY_BETWEEN	返回两个指定日期字符串之间的天数。
DAY_FROMNOW	返回当前日期与指定日期之间的天数。

---

## DAYOF 宏

DAYOF 宏仅在 IBM Campaign 中可用。

## 语法

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

## 参数

date\_string

表示有效日期的文本。

input\_format

下表中的某个关键字，用于指定 date\_string 的日期格式。

## 描述

DAYOF 以数字形式返回 date\_string 所表示日期是一个月中的第几天。如果未提供 input\_format，那么将使用缺省关键字 DELIM\_M\_D\_Y。

## 示例

DAYOF("08/31/00") 返回数字 31。

注：请参阅 第 46 页的『DATE』，以获取有效日期格式的更多相关信息。

---

## DISTANCE 宏

DISTANCE 宏在 IBM Campaign 和 IBM Interact 中都可用。

## 语法

DISTANCE(lat1, long1, lat2, long2[, UNIT\_OF\_MEASURE][, PRECISION])

## 参数

lat1

第一个点的纬度，以小数值表示。

long1

第一个点的经度，以小数值表示。

lat2

第二个点的纬度，以小数值表示。

long2

第二个点的经度，以小数值表示。

UNIT\_OF\_MEASURE

可选参数，用于指示所返回距离的计量单位。值为 MILES 或 KILOMETERS。如果省略此参数，那么将使用缺省值 MILES。

PRECISION

可选参数，用于指示所返回距离的小数点后精度级别。如果指定精度值，那么返回的距离将按您指定的小数位数截断。最大值为 5。如果省略此值，那么不会截断小数位。

## 描述

DISTANCE 计算两个点之间的距离。纬度和经度应该采用十进制单位。务必使用逗号和空格来分隔数字值。这是适应使用逗号作为小数点的语言所必需，如下面的第二个示例所示。

## 示例

DISTANCE (18.529747, 73.839798, 18.533511, 73.8777995, MILES, 2) 返回值 2.50 英里。

DISTANCE (18,529747, 73,839798, 18,533511, 73,8777995, KILOMETERS, 1) 返回值 4,0 公里。

---

## DIV 宏

DIV 宏在 IBM Campaign 和 IBM Interact 中都可用。

## 语法

```
data DIV divisor data / divisor
```

## 参数

`data`

要用作被除数的数字值。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。要了解 `data` 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

`divisor`

一个或多个值，用来除所指定数据范围内的值。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。除非 `data2` 是常量，否则 `data2` 中的列数必须与 `data1` 中的列数相等。要了解 `divisor` 的格式定义（与 `data` 相同），请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

## 描述

DIV 将指定的数据范围除以除数值。它针对每个输入列返回一个新列，其中每个新列都包含 `data1` 的相应列除以 `data2` 的相应列所得的结果（即，`data1` 的第一列除以 `data2` 的第一列，第二列除以第二列，依此类推）。

如果 `data2` 是常量，那么 `data1` 中的每个值都会除以该值。如果 `data2` 包含一个或多个列，那么将在 `data1` 中的一列与 `data2` 中的一列之间逐行执行计算。`data1` 的第一行将除以 `data2` 的第一个行值，第二行将除以第二行，依此类推。这种逐行计算会针对每一行生成一个结果，直至处理完最短列中的最后一个值为止。

注：使用每一行中的数字  $x$  都与 `divisor` 相同的列等同于使用常量  $x$  作为 `divisor`。

注：DIV 运算符可以简写为斜杠 (/)。

## 示例

TEMP = 8 DIV 4 或 TEMP = 8/4 创建名为 TEMP 的新列，其中包含值 2。
TEMP = V1/8 创建名为 TEMP 的新列，其中每个值都是 V1 列内容除以 8 所得的结果。
TEMP =V1:V3/2 创建名称分别为 TEMP、VX 和 VY 的三个新列。TEMP 列中的值是 V1 列内容除以 2 所得的结果，VX 列的值是 V2 列内容除以 2 所得的结果，而 VY 列的值是 V3 列内容除以 2 所得的结果。
TEMP = V1/V1 创建名为 TEMP 的新列，其中包含的值全为 1（因为任何数字除以自身的结果均为 1）。
TEMP = V1/V2 创建名为 TEMP 的新列，其中每个值都是列 V1 的行值除以列 V2 的相应行值所得的结果。
TEMP = V1:V3/V4:V6 创建名称分别为 TEMP、VX 和 VY 的三个新列。列 TEMP 包含 V1 中的值除以列 V4 的相应行值所得的结果。列 VX 包含列 V2 除以 V5 所得的结果。列 VY 包含列 V3 除以 V6 所得的结果。
TEMP = V1[10:20] / V2 或 TEMP = V1[10:20] / V2[1:11] 创建名为 TEMP 的新列，其中，前 11 个单元格包含 V1 列第 10-20 行的值除以列 V2 中第 1-11 行的值所得的结果。TEMP 中的其他单元格为空。

## 相关函数

函数	描述
MOD	计算所指定数据范围中内容的模数
MULT	将两个数据范围的内容相乘
POW	按指定的指数幂，计算底数值的乘幂结果

---

## EQ 宏

EQ 宏在 IBM Campaign 和 IBM Interact 中都可用。

### 语法

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

### 参数

data1

要比较的单元格范围。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

data2

要与所指定列中的所有值进行比较的数字。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。除非 data2 是常量，否则 data2 中的列数必须与 data1 中的列数相等。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

## 描述

EQ 对两个指定的数据范围进行比较，如果值相等则返回 1，否则返回 0。它针对每个输入列返回一个新列，其中每个新列都包含 data1 的相应列与 data2 的相应列进行比较所得的结果（即，data1 的第一列与 data2 的第一列进行比较，第二列与第二列进行比较，依此类推）。

如果 data2 是常量，那么 data1 中的每个值都会与该值进行比较。如果 data2 是一个列，那么将逐行执行计算。data1 中的值将与 data2 的第一行值进行比较，第二行与第二行进行比较，依此类推。这种逐行计算会针对每一行生成一个结果，直至处理完最短列中的最后一个值为止。

比较字符串时，大小写无关紧要（即，"Yes"、"YES"、"yes"和"yeS"全都视为相等）。

注：EQ 运算符可以简写为双等号 (==)。在圆括号内，也可以将单个等号 (=) 用于 EQ 宏函数（在圆括号外，等号解释为赋值运算符）。

## 示例

<pre>TEMP = 3 EQ 4、TEMP = 3==4 或 TEMP = (3=4)</pre>
创建名为 TEMP 的新列，其中包含值 0（因为 3 不等于 4）。
<pre>TEMP = "No" == "NO"</pre>
创建名为 TEMP 的新列，其中包含值 1（字符串比较不区分大小写）。
<pre>TEMP = V1 == 8</pre>
创建名为 TEMP 的新列，其中每个值都是 1（如果列 V1 的相应行值等于数字 8）或者 0。
<pre>TEMP = V1==V1</pre>
创建名为 TEMP 的新列，其中包含的值全为 1（因为每个数字都与其自身相等）。
<pre>TEMP = V1==V2</pre>
创建名为 TEMP 的新列，其中每个值都是列 V1 的行值与列 V2 的相应行值进行比较所得的结果。
<pre>TEMP = V1:V3 == V4:V6</pre>
创建名称分别为 TEMP、VX 和 VY 的三个新列。列 TEMP 包含 V1 中的值与列 V4 的相应行值进行比较所得的结果。列 VX 包含列 V2 与 V5 进行比较的结果。列 VY 包含列 V3 与 V6 进行比较的结果。
<pre>TEMP = V1[10:20] == V2 或 TEMP = V1[10:20] == V2[1:11]</pre>
创建名为 TEMP 的新列，其中，前 11 个单元格包含 V1 列第 10-20 行的值与 V2 列第 1-11 行进行比较所得的结果。TEMP 中的其他单元格为空。

## 相关函数

函数	描述
EQ	如果一个数据范围等于另一个数据范围，那么返回 TRUE
GE	如果一个数据范围大于或等于另一个数据范围，那么返回 TRUE
GT	如果一个数据范围大于另一个数据范围，那么返回 TRUE
LE	如果一个数据范围小于或等于另一个数据范围，那么返回 TRUE
LT	如果一个数据范围小于另一个数据范围，那么返回 TRUE
NE	如果一个数据范围不等于另一个数据范围，那么返回 TRUE

---

## EXP 宏

EXP 宏仅在 IBM Campaign 中可用。

### 语法

EXP(data)

### 参数

data

用作自然数  $e$  的指数的数字值。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

### 描述

EXP 以自然数  $e$  为底数，以所指定数据范围内的每个值为指数，计算乘幂结果（即，计算  $e^x$ ）。常量  $e$  等于 2.7182818。EXP 针对每个输入列返回一个新列，其中每个新列都包含相应输入列中每个  $x$  值所产生的  $e^x$ 。EXP 是 LN 宏函数的逆函数。

注：当值  $x$  过大或过小时，将会返回溢出错误。 $e^x$  超过最大或最小 32 位浮点值时就会发生这种情况。

### 示例

TEMP = EXP(2)
创建名为 TEMP 的新列，其中包含值 7.39。
TEMP = EXP(V1)
创建名为 TEMP 的新列，其中每个值都是以 $e$ 为底数并以 V1 列内容为指数的乘幂结果。

TEMP = EXP(V1:V3)
创建名称分别为 TEMP、VX 和 VY 的三个新列。TEMP 列中的值是 $e$ 以 V1 列内容为指数的乘幂结果，VX 列的值是 $e$ 以 V2 列内容为指数的乘幂结果，而 VY 列的值是 $e$ 以 V3 列内容为指数的乘幂结果。
TEMP = EXP(V1[10:20])
创建名为 TEMP 的新列，其中，前 11 个单元格包含 $e$ 以 V1 列第 10-20 行中的值为指数的乘幂结果。TEMP 中的其他单元格为空。
TEMP = EXP(V1[1:5]:V2)
创建名称分别为 TEMP 和 VX 的两个新列，每个列在第 1-5 行都有值（其他单元格为空）。列 TEMP 中的值是 $e$ 以列 V1 的相应行值为指数的乘幂结果，而列 VX 中的值是 $e$ 以列 V2 的相应行值为指数的乘幂结果。

## 相关函数

函数	描述
LN 或 LOG	计算所指定数据范围中内容的自然对数
LOG2	计算所指定数据范围中内容以 2 为底的对数
LOG10	计算所指定数据范围中内容以 10 为底的对数
POW	指数幂

## EXTERNALCALLOUT 宏

EXTERNALCALLOUT 宏仅在 IBM Interact 中可用。

### 语法

EXTERNALCALLOUT( *calloutName*, *arg1*, ...)

### 参数

*calloutName*

使用 ExternalCallout API 创建的呼出的名称。此名称必须与 IBM Marketing Platform 中创建的"外部呼出"类别的名称匹配。

*arg1*

呼出所需的自变量（如果有）。

### 描述

EXTERNALCALLOUT 使您能够调用外部应用程序，以将数据添加到交互式流程图。EXTERNALCALLOUT 可以返回您创建的呼出所返回的任何结果。您必须使用 ExternalCallout API 以 Java 编写此呼出。有关更多详细信息，请参阅《IBM Interact 管理员指南》。

## 示例

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

调用呼出 `getStockPrice`，并传递股票名称 `UNCA` 作为自变量。这个用户定义呼出会返回该呼出所定义的股票价格。

---

## FACTORIAL 宏

FACTORIAL 宏仅在 IBM Campaign 中可用。

### 语法

```
FACTORIAL(data)
```

### 参数

data

要计算阶乘的整数值。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式，但必须大于或等于零。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

### 描述

FACTORIAL 计算所指定数据范围内值的阶乘。所有输入都必须大于或等于 0 的整数。对于小于或等于 1 的整数而言，阶乘为 1。对于整数  $x \geq 2$ ，阶乘  $x! = x(x-1)(x-2) \dots(x - (x-1))$ 。FACTORIAL 针对每个输入列返回一个新列，其中每个新列都包含相应输入列中数字的阶乘。

注：任何大于 34 的值都会产生 ???（浮点溢出错误）。

### 示例

<pre>TEMP = FACTORIAL(3)</pre>
创建名为 TEMP 的新列，其中包含值 6。
<pre>TEMP = FACTORIAL(-2)</pre>
生成错误 333，指出自变量必须大于或等于 0。
<pre>TEMP = FACTORIAL(V1)</pre>
创建名为 TEMP 的新列，其中每个值都是 V1 列内容的阶乘。
<pre>TEMP = FACTORIAL(V1:V3)</pre>
创建名称分别为 TEMP、VX 和 VY 的三个新列。TEMP 列中的值是 V1 列内容的阶乘，VX 列的值是 V2 列内容的阶乘，而 VY 列的值是 V3 列内容的阶乘。
<pre>TEMP = FACTORIAL(V1[10:20])</pre>
创建名为 TEMP 的新列，其中，前 11 个单元格包含 V1 列第 10-20 行值的阶乘。TEMP 中的其他单元格为空。

```
TEMP = FACTORIAL(V1[50:99]:V2)
```

创建名称分别为 TEMP 和 VX 的两个新列，每个列在第 1-50 行都有值（其他单元格为空）。列 TEMP 中的值是列 V1 中各行的阶乘，而列 VX 中的值是列 V2 中各个值的阶乘。

## FLOOR 宏

FLOOR 宏仅在 IBM Campaign 中可用。

### 语法

```
FLOOR(data)
```

### 参数

data

要计算向下取整的数字值。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

### 描述

FLOOR 计算所指定数据范围内值的向下取整。一个数值的向下取整就是最大整数减去该数值。FLOOR 针对每个输入列返回一个新列，其中每个新列都包含相应输入列中数字的向下取整。

注：此函数与 INT 宏函数等同。

### 示例

```
TEMP = FLOOR(4.3)
```

创建名为 TEMP 的新列，其中包含值 4。

```
TEMP = FLOOR(2.9)
```

创建名为 TEMP 的新列，其中包含值 -3。

```
TEMP = FLOOR(V1)
```

创建名为 TEMP 的新列，其中每个值都是 V1 列内容的向下取整。

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

创建名称分别为 TEMP、VX 和 VY 的三个新列。TEMP 列中的值是 V1 列内容的向下取整，VX 列的值是 V2 列内容的向下取整，而 VY 列的值是 V3 列内容的向下取整。

```
TEMP = FLOOR(V1[10:20])
```

创建名为 TEMP 的新列，其中，前 11 个单元格包含 V1 列第 10-20 行值的向下取整。TEMP 中的其他单元格为空。

```
TEMP = FLOOR(V1[50:99]:V2)
```

创建名称分别为 TEMP 和 VX 的两个新列，每个列在第 1-50 行都有值（其他单元格为空）。列 TEMP 中的值是列 V1 中各行的向下取整，而列 VX 中的值是列 V2 中各个值的向下取整。

## 相关函数

函数	描述
CEILING	计算所指定数据范围内每个值的向上取整
FRACTION	返回所指定数据范围内每个值的小数部分
TRUNCATE	返回所指定数据范围内每个值的非小数部分

---

## FORMAT 宏

FORMAT 宏仅在 IBM Campaign 中可用。

### 语法

Format 有两种格式，分别用于数字数据类型和文本/字符数据类型。

对于数字数据类型：

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

对于文本/字符数据类型：

```
FORMAT(colName, width [, alignment])
```

### 参数

colName

这个宏检查 colName 并确定其数据类型，然后对后续参数相应实施适当的规则。

width

宽度应该足够大以容纳完整结果，否则结果将截断。如果 colName 是数字，可接受的宽度值为 1 到 29，否则为 1 到 255。

precision

精度是指小数点后的位数。可接受的值为 0 到 15。如果为 0，那么结果为整数。缺省精度值为 2。

format\_type

适用于 format\_type 的关键字如下所示：

PERIOD	将句点 (.) 用作小数点符号。不使用任何数字分组符号。这是缺省值。
COMMA	将逗号 (,) 用作小数点符号。不使用任何数字分组符号。
PERIOD_COMMA	将句点用作小数点符号，并将逗号用作数字分组符号。

COMMA_PERIOD	将逗号用作小数点符号，并将句点用作数字分组符号。
--------------	--------------------------

### alignment

适用于 alignment 的关键字包括 LEFT 和 RIGHT。对于数字数据类型，缺省值为 RIGHT，而对于文本/字符数据类型，缺省值为 LEFT。

### padding

适用于 padding 的关键字包括 SPACE 和 ZERO。缺省值为 SPACE。当 alignment 为 LEFT 时，将会忽略 ZERO（改为使用 SPACE）。

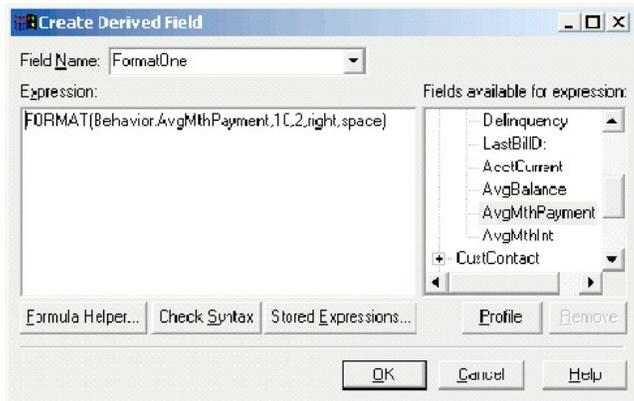
请注意，存放在文本/字符数据类型中的数字字符串视为文本/字符。另请注意，数字格式接受多个可选关键字，其每一个都有缺省值。但是，要覆盖第二个或后续可选关键字的缺省值，您必须指定前面可选关键字的缺省值（实际上，这些关键字会成为必需关键字）。例如，要将 alignment 覆盖为 LEFT，您必须指定：FORMAT(myNumCol, 10, 2, PERIOD, LEFT)。

## 描述

FORMAT 将数字数据转换为字符串格式，并可使用各种格式选项来控制 and 定义输出字符串。在创建具有特定格式的快照文件，以便通过邮件发送文件时，这个宏特别有用。

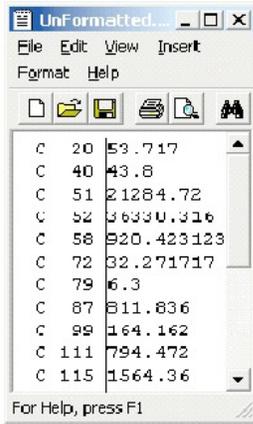
## 示例

以下示例使用 FORMAT 来定义派生字段。

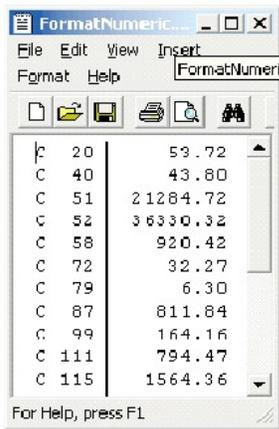


以下示例显示同一字段 AvgMthPayment 的三种格式。

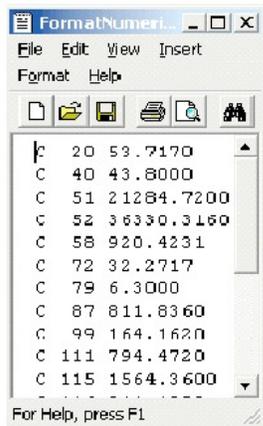
未编排格式：



使用 `FORMAT(Behavior.AvgMthPayment,10,2,right,space)` 来编排格式:



使用 `FORMAT(Behavior.AvgMthPayment,10,4)` 来编排格式:



## FRACTION 宏

FRACTION 宏仅在 IBM Campaign 中可用。

## 语法

FRACTION(data)

## 参数

data

要计算尾数的数字值。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

## 描述

FRACTION 计算所指定数据范围内值的小数部分。它针对每个输入列返回一个新列，其中每个新列都包含相应输入列中数字的小数部分。

注：FRACTION 宏函数与 TRUNCATE 宏函数互补，这两者之和等于原始值。

## 示例

TEMP = FRACTION(4.3)
创建名为 TEMP 的新列，其中包含值 0.3。
TEMP = FRACTION(2.9)
创建名为 TEMP 的新列，其中包含值 -0.9。
TEMP = FRACTION(V1)
创建名为 TEMP 的新列，其中每个值都是 V1 列内容的小数部分。
TEMP = FRACTION(V1:V3)
创建名称分别为 TEMP、VX 和 VY 的三个新列。TEMP 列中的值是 V1 列内容的小数部分，VX 列中的值是 V2 列内容的小数部分，而 VY 列中的值是 V3 列内容的小数部分。
TEMP = FRACTION(V1[10:20])
创建名为 TEMP 的新列，其中，前 11 个单元格包含 V1 列第 10-20 行值的小数部分。TEMP 中的其他单元格为空。
TEMP = FRACTION(V1[50:99]:V2)
创建名称分别为 TEMP 和 VX 的两个新列，每个列在第 1-50 行都有值（其他单元格为空）。列 TEMP 中的值是列 V1 中各行的小数部分，而列 VX 中的值是列 V2 中各个值的小数部分。

## 相关函数

函数	描述
CEILING	计算所指定数据范围内每个值的向上取整
FLOOR	计算所指定数据范围内每个值的向下取整
TRUNCATE	返回所指定数据范围内每个值的非小数部分

## GE 宏

GE 宏在 IBM Campaign 和 IBM Interact 中可用。

### 语法

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

### 参数

data1

要比较的数字单元格范围。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

data2

要与所指定列中的所有值进行比较的数字。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。除非 data2 是常量，否则 data2 中的列数必须与 data1 中的列数相等。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

### 描述

GE 对两个指定的数据范围进行比较，如果第一个数据集中的值大于或等于第二个数据集中的值，那么返回 1，否则返回 0。它针对每个输入列返回一个新列，其中每个新列都包含 data1 的相应列与 data2 的相应列进行比较所得的结果（即，data1 的第一列与 data2 的第一列进行比较，第二列与第二列进行比较，依此类推）。

如果 data2 是常量，那么 data1 中的每个值都会与该值进行比较。如果 data2 是一个列，那么将逐行执行计算。data1 中的值将与 data2 的第一行值进行比较，第二行与第二行进行比较，依此类推。这种逐行计算会针对每一行生成一个结果，直至处理完最短列中的最后一个值为止。

注：GE 运算符可以简写为大于号后跟等号 (>=)。

### 示例

```
TEMP = 9 GE 4 或 TEMP = 9 >= 4
```

创建名为 TEMP 的新列，其中包含值 1（因为 9 大于 4）。

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

创建名为 TEMP 的新列，其中每个值都是 1（如果列 V1 的相应行值大于或等于数字 8）或者 0。

```
TEMP = V1:V3 >= 2
```

创建名称分别为 TEMP、VX 和 VY 的三个新列。TEMP 列中的值是 V1 列内容与值 2 进行比较的结果，VX 列的值是 V2 列内容与值 2 进行比较的结果，而 VY 列的值是 V3 列内容与值 2 进行比较的结果。

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

创建名为 TEMP 的新列，其中包含的值全为 1（因为每个数字都与其自身相等）。

TEMP = V1 >= V2
创建名为 TEMP 的新列，其中每个值都是列 V1 的行值与列 V2 的相应行值进行比较所得的结果。
TEMP = V1:V3 >= V4:V6
创建名称分别为 TEMP、VX 和 VY 的三个新列。列 TEMP 包含 V1 中的值与列 V4 的相应行值进行比较所得的结果。列 VX 包含列 V2 与 V5 进行比较的结果。列 VY 包含列 V3 与 V6 进行比较的结果。
TEMP = V1[10:20] >= V2 或 TEMP = V1[10:20] >= V2[1:11]
创建名为 TEMP 的新列，其中，前 11 个单元格包含 V1 列第 10-20 行的值与 V2 列第 1-11 行的值进行比较所得的结果。TEMP 中的其他单元格为空。

## 相关函数

NE	如果一个数据范围不等于另一个数据范围，那么返回 TRUE
----	------------------------------

---

## GROUPBY 宏

he GROUPBY 宏仅在 IBM Campaign 中可用。

### 语法

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

### 参数

- group\_field

指定记录分组所依据的变量（即，所指定变量的所有相同值会分组到一起）。

- keyword

指定要对汇总字段执行的汇总函数。

- rolled\_field

指定要汇总的变量。

- output\_field

指定针对单一分组行所要返回的备用变量，并且只能与关键字 MinOf、MaxOf 和 MedianOf 配合使用。

### 描述

GROUPBY 对一个组内的多行数据进行汇总。此函数的输出是单个列。该输出是在 group\_field 所指定的同质组中，对 rolled\_field 执行 keyword 所指定运算的结果。如果有多个答案满足指定的条件，那么将返回所遇到的第一个答案。

如果未提供可选的 output\_field，那么输出是对 rolled\_field 执行运算的结果。如果提供 output\_field，那么结果是组中该行的 output\_field。

如果组中有多行满足指定的条件（例如，有多行的最大值相等），那么将返回与满足条件的第一行相关联的 `output-field`。

注：要针对多个列来处理分组，可以将一列字段名（以逗号分隔）括在一组花括号"`{ }`"内，并将其用作 `GROUPBY` 宏调用的第一个参数。

支持的关键字如下所示（不区分大小写）：

关键字	是否为字符串?	描述
	是/否	
CountOf	是	返回每个组中的记录数目（ <code>rolled_field</code> 可以是数字或字符串；无论 <code>rolled_field</code> 的值为何，返回的值均相同）。
MinOf	是	返回每个组中 <code>rolled_field</code> 的最小值（ <code>rolled_field</code> 可以是数字或字符串；如果 <code>rolled_field</code> 是字符串，那么将返回按字母顺序排序时最接近字母表开头的值）。
MaxOf	是	返回每个组中 <code>rolled_field</code> 的最大值（ <code>rolled_field</code> 可以是数字或字符串；如果 <code>rolled_field</code> 是字符串，那么将返回按字母顺序排序时最接近字母表末尾的值）。
DiffOf	是	返回每个组中 <code>rolled_field</code> 的不同值数目（ <code>rolled_field</code> 可以是数字或字符串）。
AvgOf	否	返回每个组中 <code>rolled_field</code> 的平均值（ <code>rolled_field</code> 必须是数字）。
ModeOf	是	返回每个组中 <code>rolled_field</code> 的众数值，即出现次数最多的值（ <code>rolled_field</code> 可以是数字或字符串）。
MedianOf	是	返回每个组中 <code>rolled_field</code> 的中值，即，按 <code>rolled_field</code> 排序时位于中间位置的值（ <code>rolled_field</code> 可以是数字或字符串；如果 <code>rolled_field</code> 是字符串，那么各个值将按字母顺序排序）。
OrderOf	是	返回每个组中 <code>rolled_field</code> 的阶（ <code>rolled_field</code> 必须是数字）。如果多个记录具有相同的值，那么它们的阶位值也相同。

关键字	是否为字符串? 是/否	描述
SumOf	否	返回每个组中 rolled_field 的总和 (rolled_field 必须是数字)。
StdevOf	否	返回每个组中 rolled_field 的标准差 (rolled_field 必须是数字)。
IndexOf	是	返回按 rolled_field 排序时, 每个记录基于 1 的索引 (rolled_field 可以是数字或字符串)。排序顺序为升序。  注: 对于数字字段, 通过在排序字段前加上减号 (-), 可以将 RankOf 和 IndexOf 的排序顺序更改为降序。
RankOf	是	返回按 rolled_field 排序时, 每个记录所属的基于 1 类别 (rolled_field 可以是数字或字符串)。排序顺序为升序。  注: 对于数字字段, 通过在排序字段前加上减号 (-), 可以将 RankOf 和 IndexOf 的排序顺序更改为降序。

## 示例

GROUPBY (Household_ID, SumOf, Account_Balance)
按家庭计算所有帐户余额的总和。
GROUPBY (Cust_ID, MinOf, Date(Account_Open_Date), Acc_Num)
返回客户开立的第一个帐户的帐号。

## GROUPBY\_WHERE 宏

GROUPBY\_WHERE 宏仅在 IBM Campaign 中可用。

### 语法

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

### 参数

- group\_field

指定记录分组所依据的变量 (即, 所指定变量的所有相同值会分组到一起)。

- keyword  
指定要执行的汇总函数。
- rolled\_field  
指定要汇总的变量。
- where\_value  
求值为值 1 或 0 的表达式，用于指定要将哪些行包括在汇总运算中。
- output\_field  
指定要针对单一分组行返回的备用变量，并且只能与关键字 MinOf、MaxOf 和 MedianOf 配合使用。

## 描述

GROUPBY\_WHERE 对一个组内的特定数据行进行汇总。此函数的输出是单个列。该输出是在 group\_field 所指定的同质组中，按 where\_value 过滤的情况下，对 rolled\_field 执行 keyword 所指定运算的结果。只有 where\_value 为 1 的行才会包括在计算中。

如果未提供可选的 output\_field，那么输出是对 rolled\_field 执行运算的结果。如果提供 output\_field，那么结果是组中该行的 output\_field。

注：请参阅第 67 页的『GROUPBY 宏』，以获取有效 keyword 值的更多相关信息。

## 示例

GROUPBY_WHERE (Household_ID, SumOf, Account_Balance, Account_Balance>0) 计算每个家庭中余额为正的所有帐户的余额总和。
GROUPBY_WHERE (Cust_ID, AvgOf, Purchase_Amt, Date(Current_Date) - Date(Purchase_Date) <90) 计算过去 90 天内每个客户的平均采购额。

---

## GT 宏

GT 宏在 IBM Campaign 和 IBM Interact 中都可用。

### 语法

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

### 参数

data1

要比较的数字单元格范围。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

data2

要与所指定列中的所有值进行比较的数字。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。除非 data2 是常量，否则 data2 中的列数必须与 data1 中的列数相等。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

## 描述

GT 对两个指定的数据范围进行比较，如果第一个数据集中的值大于第二个数据集中的值，那么返回 1，否则返回 0。它针对每个输入列返回一个新列，其中每个新列都包含 data1 的相应列与 data2 的相应列进行比较所得的结果（即，data1 的第一列与 data2 的第一列进行比较，第二列与第二列进行比较，依此类推）。

如果 data2 是常量，那么 data1 中的每个值都会与该值进行比较。如果 data2 是一个列，那么将逐行执行计算。data1 中的值将与 data2 的第一行值进行比较，第二行与第二行进行比较，依此类推。这种逐行计算会针对每一行生成一个结果，直至处理完最短列中的最后一个值为止。

注：GT 运算符可以简写为大于号 (>)。

## 示例

TEMP = 3 GT 4 或 TEMP = 3 > 4
创建名为 TEMP 的新列，其中包含值 0（因为 3 不大于 4）。
TEMP = V1 > 8
创建名为 TEMP 的新列，其中每个值都是 1（如果列 V1 的相应行值大于数字 8）或者 0。
TEMP = V1:V3 > 2
创建名称分别为 TEMP、VX 和 VY 的三个新列。TEMP 列中的值是 V1 列内容与值 2 进行比较的结果，VX 列的值是 V2 列内容与值 2 进行比较的结果，而 VY 列的值是 V3 列内容与值 2 进行比较的结果。
TEMP = V1 > V1
创建名为 TEMP 的新列，其中包含的值全为 0（因为没有任何数字会大于其自身）。
TEMP = V1 > V2
创建名为 TEMP 的新列，其中每个值都是列 V1 的行值与列 V2 的相应行值进行比较所得的结果。
TEMP = V1:V3 > V4:V6
创建名称分别为 TEMP、VX 和 VY 的三个新列。列 TEMP 包含 V1 中的值与列 V4 的相应行值进行比较所得的结果。列 VX 包含列 V2 与 V5 进行比较的结果。列 VY 包含列 V3 与 V6 进行比较的结果。
TEMP = V1[10:20] > V2 或 TEMP = V1[10:20] > V2[1:11]
创建名为 TEMP 的新列，其中，前 11 个单元格包含 V1 列第 10-20 行的值与 V2 列第 1-11 行的值进行比较所得的结果。TEMP 中的其他单元格为空。

## 相关函数

函数	描述
EQ	如果一个数据范围等于另一个数据范围，那么返回 TRUE
GE	如果一个数据范围大于或等于另一个数据范围，那么返回 TRUE
LE	如果一个数据范围小于或等于另一个数据范围，那么返回 TRUE
LT	如果一个数据范围小于另一个数据范围，那么返回 TRUE
NE	如果一个数据范围不等于另一个数据范围，那么返回 TRUE

---

## IF 宏

IF 宏在 IBM Campaign 和 IBM Interact 中都可用。

### 语法

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

### 参数

predicate\_col

布尔值列，或者求值为单个布尔值列的表达式。布尔值解释为零或非零。这个列中的行数应至少与从中抽取数据的数据范围相同。

then\_value

在 predicate\_col 的相应行包含非零值时返回的值。此参数可以是常量值、列或求值为上述任何类型的表达式。请参阅第 6 页的『适用于 IBM Campaign 的宏函数参数』，以了解 then\_value 的格式定义（与 data 相同）。

else\_value

如果提供这个可选参数，那么当 predicate\_col 的相应行包含零时，将返回此参数。此参数可以是常量值、列或求值为上述任何类型的表达式。如果未提供 else\_value，那么每当 predicate\_col 求值为 false 时，都会返回零。请参阅第 6 页的『适用于 IBM Campaign 的宏函数参数』，以了解 else\_value 的格式定义（与 data 相同）。

### 描述

IF 对 predicate\_col 中的表达式进行求值，并在该表达式为 true 时返回 then\_value，在该表达式为 false 时返回 else\_value。它在 then\_value 和 else\_value 中返回相同数量的列。新列将包含 then\_value 中相应的行值（如果 predicate\_col 的值不为零）。如果提供 else\_value，那么它会在 predicate\_col 的值为零时返回。如果未提供 else\_value，那么将返回零。

因为 IF 逐行操作，它会针对每一行生成一个结果，直至处理完最短列（即，predicate\_col、then\_value 和 else\_value 中长度最短的列）的最后一个值为止。

注：通常，您会希望使用某个比较宏函数（例如，==、>、<、ISEVEN 和 ISODD，等等）创建谓词列。

## 示例

TEMP = IF(1, V1)创建名为 TEMP 的新列，其中包含列 V1 的副本。
TEMP = IF(V1, 1, 0) 创建名为 TEMP 的新列，其中每个值都是 1（如果列 V1 的相应值不为零）或者 0。
TEMP = IF(V3, V1, V2) 创建名为 TEMP 的新列，其中每个值都是从列 V1 复制（如果列 V3 的相应值不为零），或者从列 V2 复制。
TEMP = IF(ABS(V1-AVG(V1)) < STDV(V1), V1) 创建名为 TEMP 的新列，其中包含列 V1 中每个偏离平均值不超过 1 个标准差的值。
TEMP = IF(V3[20:30], V1[30:40], V2) 创建名为 TEMP 的新列，其中包含第 10-20 行的值。每个值都是从 V1 列的第 10-20 单元格复制（如果 V3 列中第 30-40 单元格的相应值不为零），或者是从 V2 列的第 1-11 单元格复制。

---

## IN 宏

IN 宏在 IBM Campaign 和 IBM Interact 中都可用。

### 语法

valuet IN (value1 AND value2 . . . .) 或 valuet IN subquery

### 参数

第一种格式允许使用一系列值来替代子查询。

第二种格式使用子查询，该子查询将进行求值以生成中间结果，针对该结果可执行进一步的处理。

### 描述

IN 谓词允许您使用一系列值来替代子查询，或者引入子查询。

注：IN 谓词有一个否定版本，即 NOT IN。此版本的格式与 IN 完全相同。仅当在子查询所返回的值中找不到提供的值时，NOT IN 才为 true。

要点：在 IBM Interact 中使用 IN 时，只能使用 value IN (value1 AND value2 . . . .) 语法。

## 示例

TEMP = IN(25, COLUMN(1...10))
返回数据范围中的一个或多个指定列
TEMP = IN("cat", COLUMN("cat", "dog", "bird"))
创建名为 TEMP 的新列，其中包含值 1。
TEMP = IN(V1, V1)
创建名为 TEMP 的新列，其中包含的值全为 1。
TEMP = IN(V1, V2)
创建名为 TEMP 的新列，其中每个值都是 1（如果列 V1 的相应行包含列 V2 中的值）或者 0。

---

## INT 宏

INT 宏仅在 IBM Campaign 中可用。

### 语法

INT(data)

### 参数

data

要向下舍入为整数值的数字值。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

### 描述

INT 计算小于所指定数据范围内各个值的最大整数（也称为向下取整）。INT 针对每个输入列返回一个新列，其中每个新列都包含相应输入列中数字的向下取整。

注：此函数与 FLOOR 宏函数等同。

## 示例

TEMP = INT(4.7)
创建名为 TEMP 的新列，其中包含值 4。
TEMP = INT(-1.5)
创建名为 TEMP 的新列，其中包含值 -2。
TEMP = INT(V1)
创建名为 TEMP 的新列，其中每个值都是小于或等于 V1 列内容的最大整数。
TEMP = V1 - INT(V1)
创建名为 TEMP 的新列，其中包含列 V1 中每个值的小数部分。

TEMP = INT(V1:V3)
创建名称分别为 TEMP、VX 和 VY 的三个新列。TEMP 列中的值是小于或等于 V1 列内容的最大整数，VX 列的值是小于或等于 V2 列内容的最大整数，而 VY 列的值是小于或等于 V3 列内容的最大整数。
TEMP = INT(V1[10:20])
创建名为 TEMP 的新列，其中，前 11 个单元格包含小于或等于 V1 列第 10-20 行中相应值的最大整数。TEMP 中的其他单元格为空。
TEMP = INT(V1[1:5]:V2)
创建名称分别为 TEMP 和 VX 的两个新列，每个列在第 1-5 行都有值（其他单元格为空）。列 TEMP 中的值是小于或等于列 V1 中相应行值的最大整数，而列 VX 中的值是小于或等于列 V2 中相应行值的最大整数。

## 相关函数

函数	描述
ROUND	计算所指定数据范围中内容的取整值
TRUNCATE	返回所指定数据范围内每个值的非小数部分

---

## INVERSE 宏

INVERSE 宏仅在 IBM Campaign 中可用。

### 语法

INVERSE(data)

### 参数

data

要计算相反数的数字值。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

### 描述

INVERSE 计算所指定数据范围内值的相反数。它返回  $-x$ （即，负值作为正值返回，而正值作为负值返回）。INVERSE 针对每个输入列返回一个新列，其中每个新列都包含相应输入列中值的相反数。

注：要反转某个值或列，请在它前面加上减号 (-)。例如， $V2 = -V1$  等同于  $V2 = \text{INVERSE}(V1)$ 。

### 示例

TEMP = INVERSE(3.2)
创建名为 TEMP 的新列，其中包含值 -3.2。

TEMP = INVERSE(V1)
创建名为 TEMP 的新列，其中每个值都是 V1 列中值的相反数。
TEMP = INVERSE(V1:V3)
创建名称分别为 TEMP、VX 和 VY 的三个新列。TEMP 列中的值是 V1 列中值的相反数，VX 列的值是 V2 列中值的相反数，而 VY 列的值是 V3 列中值的相反数。
TEMP = INVERSE(V1[10:20])
创建名为 TEMP 的新列，其中，前 11 个单元格包含 V1 列第 10-20 行值的相反数。TEMP 中的其他单元格为空。
TEMP = INVERSE(V1[1:5]:V2)
创建名称分别为 TEMP 和 VX 的两个新列，每个列在第 1-5 行都有值（其他单元格为空）。列 TEMP 中的值是列 V1 中相应行值的相反数，而列 VX 中的值是列 V2 中相应行值的相反数。

## 相关函数

函数	描述
ABS	计算所指定数据范围中内容的绝对值
NOT	计算所指定数据范围中内容的逻辑 NOT 运算结果
SIGN	计算所指定数据范围内值的符号（正或负）

---

## IS 宏

IS 宏在 IBM Campaign 和 IBM Interact 中都可用。

### 语法

IS <keyword>

### 参数

keyword

搜索条件，通常为"NULL"、"TRUE"、"UNKNOWN"和"FALSE"。

### 描述

IS 用于复杂搜索条件。搜索越复杂，IS 条件就越有用。这些布尔搜索条件提供了表达基本搜索条件的替代方法。

在 IBM Interact 与 IBM Campaign 中，IS 返回不同的结果。NULL 在至少一个受众标识为 NULL 值时返回 1。UNKNOWN 在受众标识没有任何值时返回 1。

---

## ISERROR 宏

ISERROR 宏仅在 IBM Campaign 中可用。

## 语法

ISERROR(data)

## 参数

data

要进行检验，以确定其中是否有任何行包含错误（即，??? 单元格）的值。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

## 描述

ISERROR 检查所指定数据范围的每一行，确定是否有任何单元格包含错误（即，??? 单元格）。它返回一个新列，其中的每一行都在 data 的相应行包含错误时包含 1。否则，该行将包含 0。这种逐行计算会针对每一行生成一个结果，直至处理完最长列中的最后一个值为止。

## 示例

TEMP = ISERROR(-3) 创建名为 TEMP 的新列，其中包含值 0。
TEMP = ISERROR(V1) 创建名为 TEMP 的新列，其中每个值都是 1（如果列 V1 的相应行包含 ???）或者 0。
TEMP = ISERROR(V1:V3) 创建名为 TEMP 的新列，其中每个值都是 1（如果列 V1 - V3 中相应行的任何单元格包含 ???）或者 0。
TEMP = ISERROR(V1[50:100]:V10) 创建名为 TEMP 的新列，其中的 1-50 行包含值。每个值都是 1（如果列 V1 - V10 中第 50-100 行的任何单元格包含 ???）或 0。

---

## ISODD 宏

ISODD 宏仅在 IBM Campaign 中可用。

## 语法

ISODD(data)

## 参数

data

要检验是否为奇数的数字值。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

## 描述

ISODD 检验所指定数据集中的每个值是否为奇数。它针对每个输入列返回一个新列，其中每个新列都会针对所有奇数值（即，值以 2 取模的结果为 1）包含 1，或针对所有非奇数值（即，偶数值）包含 0。

注：对于非整数值，将先应用宏函数 INT。例如，ISODD(2.5) = 0，因为 2 不是奇数。

## 示例

TEMP = ISODD(-3)
创建名为 TEMP 的新列，其中包含值 1。
TEMP = ISODD(V1)
创建名为 TEMP 的新列，其中每个值都是检验 V1 列内容是否为奇数的结果。
TEMP = ISODD(V1:V3)
创建名称分别为 TEMP、VX 和 VY 的三个新列。TEMP 列中的值是检验 V1 列内容是否为奇数的结果，VX 列的值是检验 V2 列内容是否为奇数的结果，而 VY 列的值是检验 V3 列内容是否为奇数的结果。
TEMP = ISODD(V1[10:20])
创建名为 TEMP 的新列，其中，前 11 个单元格包含检验 V1 列第 10-20 行中的值是否为奇数的结果。TEMP 中的其他单元格为空。
TEMP = ISODD(V1[1:5]:V2)
创建名称分别为 TEMP 和 VX 的两个新列，每个列在第 1-5 行都有值（其他单元格为空）。列 TEMP 中的值是检验列 V1 中的相应行是否为奇数的结果，而列 VX 中的值是检验列 V2 中的相应行是否为奇数的结果。

## 相关函数

函数	描述
ISEVEN	检验输入值是否为偶数（即，可被 2 整除）

---

## ISEVEN 宏

ISEVEN 宏仅在 IBM Campaign 中可用。

### 语法

ISEVEN(data)

### 参数

data

要检验是否为偶数的数字值。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

## 描述

ISEVEN 检验所指定数据集中的每个值是否为偶数。它针对每个输入列返回一个新列，其中每个新列都会针对所有偶数值（即，值以 2 取模的结果为 0）包含 1，或针对所有非偶数值（即，奇数值）包含 0。

注：对于非整数值，将先应用宏函数 INT。例如，ISEVEN(2.5) = 1，因为 2 是偶数。

## 示例

TEMP = ISEVEN(-3)
创建名为 TEMP 的新列，其中包含值 0。
TEMP = ISEVEN(V1)
创建名为 TEMP 的新列，其中每个值都是检验 V1 列内容是否为偶数的结果。
TEMP = ISEVEN(V1:V3)
创建名称分别为 TEMP、VX 和 VY 的三个新列。TEMP 列中的值是检验 V1 列内容是否为偶数的结果，VX 列的值是检验 V2 列内容是否为偶数的结果，而 VY 列的值是检验 V3 列内容是否为偶数的结果。
TEMP = ISEVEN(V1[10:20])
创建名为 TEMP 的新列，其中，前 11 个单元格包含检验 V1 列第 10-20 行中的值是否为偶数的结果。TEMP 中的其他单元格为空。
TEMP = ISEVEN(V1[1:5]:V2)
创建名称分别为 TEMP 和 VX 的两个新列，每个列在第 1-5 行都有值（其他单元格为空）。列 TEMP 中的值是检验列 V1 中的相应行是否为偶数的结果，而列 VX 中的值是检验列 V2 中的相应行是否为偶数的结果。

## 相关函数

函数	描述
ISODD	检验输入值是否为奇数（即，不可被 2 整除）

---

## ISODD 宏

ISODD 宏仅在 IBM Campaign 中可用。

### 语法

ISODD(data)

### 参数

data

要检验是否为奇数的数字值。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

## 描述

ISODD 检验所指定数据集中的每个值是否为奇数。它针对每个输入列返回一个新列，其中每个新列都会针对所有奇数值（即，值以 2 取模的结果为 1）包含 1，或针对所有非奇数值（即，偶数值）包含 0。

注：对于非整数值，将先应用宏函数 INT。例如，ISODD(2.5) = 0，因为 2 不是奇数。

## 示例

TEMP = ISODD(-3)
创建名为 TEMP 的新列，其中包含值 1。
TEMP = ISODD(V1)
创建名为 TEMP 的新列，其中每个值都是检验 V1 列内容是否为奇数的结果。
TEMP = ISODD(V1:V3)
创建名称分别为 TEMP、VX 和 VY 的三个新列。TEMP 列中的值是检验 V1 列内容是否为奇数的结果，VX 列的值是检验 V2 列内容是否为奇数的结果，而 VY 列的值是检验 V3 列内容是否为奇数的结果。
TEMP = ISODD(V1[10:20])
创建名为 TEMP 的新列，其中，前 11 个单元格包含检验 V1 列第 10-20 行中的值是否为奇数的结果。TEMP 中的其他单元格为空。
TEMP = ISODD(V1[1:5]:V2)
创建名称分别为 TEMP 和 VX 的两个新列，每个列在第 1-5 行都有值（其他单元格为空）。列 TEMP 中的值是检验列 V1 中的相应行是否为奇数的结果，而列 VX 中的值是检验列 V2 中的相应行是否为奇数的结果。

## 相关函数

函数	描述
ISEVEN	检验输入值是否为偶数（即，可被 2 整除）

---

## LE 宏

LE 宏在 IBM Campaign 和 IBM Interact 中都可用。

### 语法

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

### 参数

data1

要比较的数字单元格范围。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

data2

要与所指定列中的所有值进行比较的数字。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

## 描述

LE 对两个指定的数据范围进行比较，如果第一个数据集中的值小于或等于第二个数据集中的值，那么返回 1，否则返回 0。它针对每个输入列返回一个新列，其中每个新列都包含 data1 的相应列与 data2 的相应列进行比较所得的结果（即，data1 的第一列与 data2 的第一列进行比较，第二列与第二列进行比较，依此类推）。

如果 data2 是常量，那么 data1 中的每个值都会与该值进行比较。如果 data2 是一个列，那么将逐行执行计算。data1 中的值将与 data2 的第一行值进行比较，第二行与第二行进行比较，依此类推。这种逐行计算会针对每一行生成一个结果，直至处理完最短列中的最后一个值为止。

注：LE 运算符可以简写为小于号后跟等号 (<=)。

## 示例

TEMP = 4 LE 4 或 TEMP = 4 <= 4
创建名为 TEMP 的新列，其中包含值 1（因为 4 等于其自身）。
TEMP = V1 <= 8
创建名为 TEMP 的新列，其中每个值都是 1（如果列 V1 的相应行值小于或等于数字 8）或者 0。
TEMP = V1:V3 <= 2
创建名称分别为 TEMP、VX 和 VY 的三个新列。TEMP 列中的值是 V1 列内容与值 2 进行比较的结果，VX 列的值是 V2 列内容与值 2 进行比较的结果，而 VY 列的值是 V3 列内容与值 2 进行比较的结果。
TEMP = V1 <= V1
创建名为 TEMP 的新列，其中包含的值全为 1（因为每个数字都与其自身相等）。
TEMP = V1 <= V2
创建名为 TEMP 的新列，其中每个值都是列 V1 的行值与列 V2 的相应行值进行比较所得的结果。
TEMP = V1[10:20] <= V2 或 TEMP = V1[10:20] <= V2[1:11]
创建名为 TEMP 的新列，其中，前 11 个单元格包含 V1 列第 10-20 行的值与 V2 列第 1-11 行的值进行比较所得的结果。TEMP 中的其他单元格为空。

## 相关函数

函数	描述
EQ	如果一个数据范围等于另一个数据范围，那么返回 TRUE
GE	如果一个数据范围大于或等于另一个数据范围，那么返回 TRUE

函数	描述
GT	如果一个数据范围大于另一个数据范围，那么返回 TRUE
LT	如果一个数据范围小于另一个数据范围，那么返回 TRUE
NE	如果一个数据范围不等于另一个数据范围，那么返回 TRUE

## LIKE 宏

LIKE 宏在 IBM Campaign 和 IBM Interact 中都可用。

### 语法

data1 [NOT] LIKE data2

### 参数

data1

要比较的单元格范围。此参数可以是文本字符串，或者是求值为文本字符串的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

data2

要与所指定列中的所有值进行比较的文本模式。此参数可以是文本字符串，或者是求值为文本字符串的表达式。除非 data2 是常量，否则 data2 中的列数必须与 data1 中的列数相等。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

data2 中的下划线 ( \_ ) 表示将与 data1 中的任意单个字符匹配的通配符。百分号 ( % ) 将与 data1 中的零个或零个以上字符匹配。

### 描述

LIKE 对两个指定的数据范围进行比较，如果字符串匹配则返回 1，否则返回 0。它针对每个输入列返回一个新列，其中每个新列都包含 data1 的相应列与 data2 的相应列进行比较所得的结果（即，data1 的第一列与 data2 的第一列进行比较，第二列与第二列进行比较，依此类推）。

如果 data2 是字符串常量，那么 data1 中的每个字符串都会与该字符串进行比较。如果 data2 是一个列，那么将逐行执行计算。data1 中的第一行字符串将与 data2 的第一行字符串进行比较，第二行与第二行进行比较，依此类推。这种逐行计算会针对每一行生成一个结果，直至处理完最短列中的最后一个字符串为止。

比较字符串时，大小写无关紧要（即，"Yes"、"YES"、"yes"和"yeS"全都视为相等）。

注：LIKE 宏有一个否定版本，即 NOT LIKE。此版本的格式与 LIKE 完全相同。NOT LIKE 在 data1 中的字符串与 data2 所定义的模板不匹配时返回 1。

## 示例

TEMP = "gold" LIKE "gold"
创建名为 TEMP 的新列，其中包含值 1（因为两个字符串匹配）。
TEMP = "No" LIKE "NO"
创建名为 TEMP 的新列，其中包含值 1（字符串比较不区分大小写）。
TEMP = V1 LIKE "gold%"
创建名为 TEMP 的新列，其中每个值都是 1（如果列 V1 的相应行值等于字符串"gold"后跟任意数目的字符）或 0。
TEMP = V1 LIKE "g_ld"
创建名为 TEMP 的新列，其中每个值都是 1（如果列 V1 的相应行值等于字符串"g"后跟任意数目，再后跟"ld"）或 0。
TEMP = V1 LIKE V1
创建名为 TEMP 的新列，其中包含的值全为 1（因为每个数字都与其自身相等）。
TEMP = V1 LIKE V2
创建名为 TEMP 的新列，其中每个值都是列 V1 的行值与列 V2 的相应行值进行比较所得的结果。
TEMP = V1:V3 LIKE V4:V6
创建名称分别为 TEMP、VX 和 VY 的三个新列。列 TEMP 包含 V1 中的字符串与列 V4 的相应行字符串进行比较所得的结果。列 VX 包含列 V2 与 V5 进行比较的结果。列 VY 包含列 V3 与 V6 进行比较的结果。
TEMP = V1[10:20] LIKE V2 或 TEMP = V1[10:20] LIKE V2[1:11]
创建名为 TEMP 的新列，其中，前 11 个单元格包含 V1 列第 10-20 行的字符串与 V2 列第 1-11 行进行比较所得的结果。TEMP 中的其他单元格为空。

## 相关函数

函数	描述
EQ	如果一个数据范围等于另一个数据范围，那么返回 TRUE
GE	如果一个数据范围大于或等于另一个数据范围，那么返回 TRUE
GT	如果一个数据范围大于另一个数据范围，那么返回 TRUE
LE	如果一个数据范围小于或等于另一个数据范围，那么返回 TRUE
LT	如果一个数据范围小于另一个数据范围，那么返回 TRUE
NE	如果一个数据范围不等于另一个数据范围，那么返回 TRUE

## LN 或 LOG 宏

LN 或 LOG 宏仅在 IBM Campaign 中可用。

### 语法

LN(data) 或 LOG(data)

### 参数

data

要计算自然对数的数字值。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

### 描述

LN 或 LOG 计算所指定数据范围内每个值的自然对数。它针对每个输入列返回一个新列，其中每个新列都包含相应输入列中数字的自然对数。自然对数以常量  $e$  (2.7182818) 作为底数。LN 是 EXP 宏函数的逆函数。

注：所指定数据范围中的所有值都必须大于零。否则，对于每个无效输入，将返回空白单元格。

### 示例

TEMP = LN(3) 或 TEMP = LOG(3)
创建名为 TEMP 的新列，其中包含值 1.099。
TEMP = LN(V1)
创建名为 TEMP 的新列，其中每个值都是 V1 列内容的自然对数。
TEMP = LN(V1:V3)
创建名称分别为 TEMP、VX 和 VY 的三个新列。TEMP 列中的值是 V1 列内容的自然对数，VX 列的值是 V2 列内容的自然对数，而 VY 列的值是 V3 列内容的自然对数。
TEMP = LN(V1[10:20])
创建名为 TEMP 的新列，其中，前 11 个单元格包含 V1 列第 10-20 行值的自然对数。TEMP 中的其他单元格为空。
TEMP = LN(V1[1:5]:V2)
创建名称分别为 TEMP 和 VX 的两个新列，每个列在第 1-5 行都有值（其他单元格为空）。列 TEMP 中的值是列 V1 中相应行的自然对数，而列 VX 中的值是列 V2 中相应行的自然对数。

### 相关函数

函数	描述
EXP	以自然数 (e) 为底数，并以所指定数据范围中的每个单元格内容为指数，计算乘幂结果
LOG2	计算所指定数据范围中内容以 2 为底的对数

函数	描述
LOG10	计算所指定数据范围中内容以 10 为底的对数
POW	按指定的指数幂，计算底数值的乘幂结果

## LOG2 宏

LOG2 宏仅在 IBM Campaign 中可用。

### 语法

LOG2(data)

### 参数

data

要计算以 2 为底对数的数字值。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

### 描述

LOG2 计算所指定数据范围内值的以 2 为底对数。它针对每个输入列返回一个新列，其中每个新列都包含相应输入列中数字的以 2 为底的对数。

注：所指定数据范围中的所有值都必须大于零。否则，对于每个无效输入，将返回空白单元格。

### 示例

TEMP = LOG2(8)
创建名为 TEMP 的新列，其中包含值 3。
TEMP = LOG2(V1)
创建名为 TEMP 的新列，其中每个值都是 V1 列内容的以 2 为底对数。
TEMP = LOG2(V1:V3)
创建名称分别为 TEMP、VX 和 VY 的三个新列。TEMP 列中的值是 V1 列内容的以 2 为底对数，VX 列的值是 V2 列内容的以 2 为底对数，而 VY 列的值是 V3 列内容的以 2 为底对数。
TEMP = LOG2(V1[10:20])
创建名为 TEMP 的新列，其中，前 11 个单元格包含 V1 列第 10-20 行值的以 2 为底对数。TEMP 中的其他单元格为空。
TEMP = LOG2(V1[1:5]:V2)
创建名称分别为 TEMP 和 VX 的两个新列，每个列在第 1-5 行都有值（其他单元格为空）。列 TEMP 中的值是列 V1 中相应行的以 2 为底对数，而列 VX 中的值是列 V2 中相应行的以 2 为底对数。

## 相关函数

函数	描述
LN 或 LOG	计算所指定数据范围中内容的自然对数
LOG10	计算所指定数据范围中内容以 10 为底的对数
POW	指数幂

---

## LOG10 宏

LOG10 宏仅在 IBM Campaign 中可用。

### 语法

LOG10(data)

### 参数

data

要计算以 10 为底对数的数字值。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

### 描述

LOG10 计算所指定数据范围内值的以 10 为底对数。它针对每个输入列返回一个新列，其中每个新列都包含相应输入列中数字的以 10 为底对数。

注：所指定数据范围中的所有值都必须大于零。否则，对于每个无效输入，将返回空白单元格。

### 示例

TEMP = LOG10(100)
创建名为 TEMP 的新列，其中包含值 2。
TEMP = LOG10(V1)
创建名为 TEMP 的新列，其中每个值都是 V1 列内容的以 10 为底对数。
TEMP = LOG10(V1:V3)
创建名称分别为 TEMP、VX 和 VY 的三个新列。TEMP 列中的值是 V1 列内容的以 10 为底对数，VX 列的值是 V2 列内容的以 10 为底对数，而 VY 列的值是 V3 列内容的以 10 为底对数。
TEMP = LOG10(V1[10:20])
创建名为 TEMP 的新列，其中，前 11 个单元格包含 V1 列第 10-20 行值的以 10 为底对数。TEMP 中的其他单元格为空。
TEMP = LOG10(V1[1:5]:V2)
创建名称分别为 TEMP 和 VX 的两个新列，每个列在第 1-5 行都有值（其他单元格为空）。列 TEMP 中的值是列 V1 中相应行的以 10 为底对数，而列 VX 中的值是列 V2 中相应行的以 10 为底对数。

## 相关函数

函数	描述
LN 或 LOG	计算所指定数据范围中内容的自然对数
LOG2	计算所指定数据范围中内容以 2 为底的对数
POW	指数幂

---

## LOWER 宏

LOWER 宏在 IBM Campaign 和 IBM Interact 中都可用。

### 语法

LOWER(data)

### 参数

data

要转换为小写的字符串值。

### 描述

LOWER 将所指定数据范围内的每个字符串值转换为小写。它返回一个新列，其中每个单元格都包含相应输入单元格的小写字符串。

### 示例

Temp = LOWER "GOLD"
创建名为 Temp 的新列，其中包含 "gold"。
TEMP = LOWER( "JAN 15, 1997")
创建名为 TEMP 的新列，其中包含 ASCII 文本字符串 "jan 15, 1997"。
TEMP = LOWER( "Pressure")
创建名为 TEMP 的新列，其中包含 ASCII 文本字符串 "pressure"。
TEMP = LOWER(V1)
创建名为 TEMP 的新列，其中包含列 V1 中每个字符串的小写字符。

---

## LT 宏

LT 宏在 IBM Campaign 和 IBM Interact 中都可用。

### 语法

data1 LT data2 data1 < data2

## 参数

data1

要比较的数字单元格范围。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

data2

要与所指定列中的所有值进行比较的数字。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

## 描述

LT 对两个指定的数据范围进行比较，如果第一个数据集中的值小于第二个数据集中的值，那么返回 1，否则返回 0。它针对每个输入列返回一个新列，其中每个新列都包含 data1 的相应列与 data2 的相应列进行比较所得的结果（即，data1 的第一列与 data2 的第一列进行比较，第二列与第二列进行比较，依此类推）。

如果 data2 是常量，那么 data1 中的每个值都会与该值进行比较。如果 data2 是一个列，那么将逐行执行计算。data1 中的值将与 data2 的第一行值进行比较，第二行与第二行进行比较，依此类推。这种逐行计算会针对每一行生成一个结果，直至处理完最短列中的最后一个值为止。

注：LT 运算符可以简写为小于号 (<)。

## 示例

TEMP = 3 LT 4 或 TEMP = 3 < 4
创建名为 TEMP 的新列，其中包含值 1（因为 3 小于 4）。
TEMP = V1 < 8
创建名为 TEMP 的新列，其中每个值都是 1（如果列 V1 的相应行值小于数字 8）或者 0。
TEMP = V1:V3 < 2
创建名称分别为 TEMP、VX 和 VY 的三个新列。TEMP 列中的值是 V1 列内容与值 2 进行比较的结果，VX 列的值是 V2 列内容与值 2 进行比较的结果，而 VY 列的值是 V3 列内容与值 2 进行比较的结果。
TEMP = V1 < V1
创建名为 TEMP 的新列，其中包含的值全为 0（因为没有任何数字会小于其自身）。
TEMP = V1 < V2
创建名为 TEMP 的新列，其中每个值都是列 V1 的行值与列 V2 的相应行值进行比较所得的结果。
TEMP = V1[10:20] < V2 或 TEMP = V1[10:20] < V2[1:11]
创建名为 TEMP 的新列，其中，前 11 个单元格包含 V1 列第 10-20 行的值与 V2 列第 1-11 行进行比较所得的结果。TEMP 中的其他单元格为空。

## 相关函数

函数	描述
EQ	如果一个数据范围等于另一个数据范围，那么返回 TRUE
GE	如果一个数据范围大于或等于另一个数据范围，那么返回 TRUE
GT	如果一个数据范围大于另一个数据范围，那么返回 TRUE
LE	如果一个数据范围小于或等于另一个数据范围，那么返回 TRUE

---

## LTRIM 宏

LTRIM 宏在 IBM Campaign 和 IBM Interact 中都可用。

### 语法

LTRIM(data)

### 参数

data

要从中除去前导空格的字符串。

### 描述

LTRIM 从所指定数据范围的每个字符串值中除去前导空格字符，并返回转换后的字符串。它针对每个输入列返回一个新列。

### 示例

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

创建名为 Temp 的新字符串，其中包含 "gold"。

---

## MAX 宏

MAX 宏在 IBM Campaign 和 IBM Interact 中都可用。

### 语法

MAX(data [, keyword])

### 参数

data

要计算最大值的数字值。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

keyword

这个可选关键字确定如何对输入数据范围执行计算。请选择下列其中一项：

ALL - 对 data 中的所有单元格执行计算（缺省值）

COL - 对 data 的每一列单独执行计算

ROW - 对 data 的每一行单独执行计算

有关在 IBM Campaign 中使用关键字的更多详细信息，请参阅第 6 页的『格式规范』。

注：许多宏函数使用关键字参数 {ALL | COL | ROW}。这些关键字在 IBM Campaign 中不适用，因为输入数据始终为单个列或字段。这个宏的行为始终像是已指定 COL 关键字一样。因此，使用 IBM Campaign 时，无需指定这些关键字。

## 描述

MAX 计算所指定数据范围内值的最大值。它返回单个新列，其中包含最大值。

## 示例

TEMP = MAX(3) 或 TEMP = MAX(3, ALL)
创建名为 TEMP 的新列，其中包含值 3。
TEMP = MAX(V1)
创建名为 TEMP 的新列，其中包含单个值，即 V1 列内容的最大值。
TEMP = MAX(V1:V3)
创建名为 TEMP 的新列，其中包含单个值，即 V1、V2 和 V3 列的最大值。
TEMP = MAX(V1[10:20])
创建名为 TEMP 的新列，其中包含单个值，即 V1 列第 10-20 行单元格的值。
TEMP = MAX(V1[1:5]:V4)
创建名为 TEMP 的新列，其中包含单个值，即 V1 至 V4 列第 1-5 行单元格的值。
TEMP = MAX(V1:V3, COL)
创建名称分别为 TEMP、VX 和 VY 的三个新列。TEMP 列中的单个值是 V1 列内容的最大值，VX 列中的单个值是 V2 列内容的最大值，而 VY 列中的单个值是 V3 列内容的最大值。
TEMP = MAX(V1[1:5]:V3, COL)
创建名称分别为 TEMP、VX 和 VY 的三个新列，其中每个列都包含单个值。TEMP 列中的值是 V1 列第 1-5 行单元格的值，VX 列中的值是 V2 列第 1-5 行单元格的值，而 VY 列中的值是 V3 列第 1-5 行单元格的值。
TEMP = MAX(V1:V3, ROW)
创建名为 TEMP 的新列，其中每个单元格条目都是 V1、V2 和 V3 列中相应行的最大值。

```
TEMP = MAX(V1[10:20]:V3, ROW)
```

创建名为 TEMP 的新列，其中，前 11 个单元格包含 V1 到 V3 列第 10-20 行值的最大值。TEMP 中的其他单元格为空。

## 相关函数

函数	描述
MIN	计算一系列单元格的最小值

---

## MEAN 宏

MEAN 宏在 IBM Campaign 和 IBM Interact 中都可用。

### 语法

MEAN(data [, keyword])

### 参数

data

要计算算术平均值的数字值。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

keyword

这个可选关键字确定如何对输入数据范围执行计算。请选择下列其中一项：

ALL - 对 data 中的所有单元格执行计算（缺省值）

COL - 对 data 的每一列单独执行计算

ROW - 对 data 的每一行单独执行计算

有关使用关键字的详细信息，请参阅第 46 页的『DATE』。

注：许多宏函数使用关键字参数 {ALL | COL | ROW}。这些关键字在 IBM Campaign 中不适用，因为输入数据始终为单个列或字段。这个宏的行为始终像是已指定 COL 关键字一样。因此，使用 IBM Campaign 时，无需指定这些关键字。

### 描述

MEAN 计算所指定数据范围内单元格的平均值或算术平均值。算术平均值的计算方法是，将所有单元格内容的总和除以单元格数。MEAN 所返回的列数取决于 keyword。

- 如果 keyword 为 ALL，那么 MEAN 将返回一个新列，其中包含单个值（data 中所有单元格的平均值）。
- 如果 keyword 为 COL，那么 MEAN 将针对每个输入列返回一个新列。每个新列都包含一个值（相应输入列中所有单元格的平均值）。

- 如果 keyword 为 ROW，那么 MEAN 将返回一个新列，其中包含 data 中每一行的平均值。

注：平均数将会忽略空白单元格。

注：MEAN 与 AVG 宏函数等同。

## 示例

TEMP = MEAN(V1)
创建名为 TEMP 的新列，其中包含单个值，即 V1 列内容的算术平均值。
TEMP = MEAN(V1:V3)
创建名为 TEMP 的新列，其中包含单个值，即 V1、V2 和 V3 列内容的算术平均值。
TEMP = MEAN(V1[10:20])
创建名为 TEMP 的新列，其中包含单个值，即 V1 列第 10-20 行单元格的算术平均值。
TEMP = MEAN(V1[1:5]:V4)
创建名为 TEMP 的新列，其中包含单个值，即 V1 至 V4 列第 1-5 行单元格的算术平均值。
TEMP = MEAN(V1:V3, COL)
创建名称分别为 TEMP、VX 和 VY 的三个新列。TEMP 列中的单个值是 V1 列内容的算术平均值，VX 列中的单个值是 V2 列内容的算术平均值，而 VY 列中的单个值是 V3 列内容的算术平均值。
TEMP = MEAN(V1[10:20]:V3, COL)
创建名称分别为 TEMP、VX 和 VY 的三个新列，其中每个列都包含单个值。TEMP 列中的值是 V1 列第 10-20 行单元格的算术平均值，VX 列中的值是 V2 列第 10-20 行单元格的算术平均值，而 VY 列中的值是 V3 列第 10-20 行单元格的算术平均值。
TEMP = MEAN(V1:V3, ROW)
创建名为 TEMP 的新列，其中每个单元格条目都是 V1、V2 和 V3 列中相应行的算术平均值。
TEMP = MEAN(V1[1:5]:V3,ROW)
创建名为 TEMP 的新列，其中 1-5 行的单元格包含 V1 到 V3 列中相应行的算术平均值。TEMP 中的其他单元格为空。

## 相关函数

函数	描述
SUM 或 TOTAL	计算某个范围的单元格的总和

## MIN 宏

MIN 宏在 IBM Campaign 和 IBM Interact 中可用。

### 语法

MIN(data [, keyword])

## 参数

data

要计算最小值的数字值。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

keyword

这个可选关键字确定如何对输入数据范围执行计算。请选择下列其中一项：

ALL - 对 data 中的所有单元格执行计算（缺省值）

COL - 对 data 的每一列单独执行计算

ROW - 对 data 的每一行单独执行计算

有关使用关键字的详细信息，请参阅第 46 页的『DATE』。

注：许多宏函数使用关键字参数 {ALL | COL | ROW}。这些关键字在 **IBM Campaign** 中不适用，因为输入数据始终为单个列或字段。这个宏的行为始终像是已指定 COL 关键字一样。因此，使用 **IBM Campaign** 时，无需指定这些关键字。

## 描述

MIN 计算所指定数据范围内所有单元格的最小值。它返回单个列，其中包含最小值。

## 示例

TEMP = MIN(V1)
创建名为 TEMP 的新列，其中包含单个值，即 V1 列的最小值。
TEMP = MIN(V1:V3)
创建名为 TEMP 的新列，其中包含单个值，即 V1、V2 和 V3 列的最小值。
TEMP = MIN(V1[10:20])
创建名为 TEMP 的新列，其中包含单个值，即 V1 列第 10-20 行单元格的最小值。
TEMP = MIN(V1[1:5]:V4)
创建名为 TEMP 的新列，其中包含单个值，即 V1 至 V4 列第 1-5 行单元格的最小值。
TEMP = MIN(V1:V3, COL)
创建名称分别为 TEMP、VX 和 VY 的三个新列。TEMP 列中的单个值是 V1 列的最小值，VX 列中的单个值是 V2 列的最小值，而 VY 列中的单个值是 V3 列的最小值。
TEMP = MIN(V1[1:5]:V3, COL)
创建名称分别为 TEMP、VX 和 VY 的三个新列，其中每个列都包含单个值。TEMP 列中的值是 V1 列第 1-5 行单元格的最小值，VX 列中的值是 V2 列第 1-5 行单元格的最小值，而 VY 列中的值是 V3 列第 1-5 行单元格的最小值。
TEMP = MIN(V1:V3, ROW)
创建名为 TEMP 的新列，其中每个单元格条目都是 V1、V2 和 V3 列中相应行的最小值。

```
TEMP = MIN(V1[10:20]:V3, ROW)
```

创建名为 TEMP 的新列，其中，前 11 个单元格包含 V1 到 V3 列第 10-20 行值的最小值。TEMP 中的其他单元格为空。

## 相关函数

函数	描述
MAX	计算一系列单元格的最大值
MAX_TO_INDEX	返回所指定列的每一行中最大值的列索引

---

## MINUS 宏

MINUS 宏在 IBM Campaign 和 IBM Interact 中都可用。

### 语法

```
data MINUS subtrahend data - subtrahend
```

### 参数

data

包含被减数的单元格范围。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

subtrahend

要从所指定列的所有值中减去的数值。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。除非 subtrahend 是常量，否则 subtrahend 中的列数必须与 data 中的列数相等。要了解 subtrahend 的格式定义（与 data 相同），请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

### 描述

MINUS 从指定的数据范围 data 中减去 subtrahend。它针对每个输入列返回一个新列，其中每个新列都包含 data 的相应列减去 subtrahend 的相应列所得的结果（即，data 的第一列减去 subtrahend 的第一列，第二列减去第二列，依此类推）。

如果 subtrahend 是常量，那么 data 中的每个值都会减去该值。如果 subtrahend 包含一个或多个列，那么将在 data 中的一列与 subtrahend 中的一列之间逐行执行计算。data 的第一行将减去 subtrahend 的第一行值，第二行减去第二行，依此类推。这种逐行计算会针对每一行生成一个结果，直至处理完最短列中的最后一个值为止。

注：MINUS 运算符可以简写为减号或连字符 (-)。

## 示例

TEMP = 7 MINUS 4 或 TEMP = 7 - 4 创建名为 TEMP 的新列，其中包含值 3。
TEMP = V1 - 8 创建名为 TEMP 的新列，其中每个值都是 V1 列内容减去 8 所得的结果。
TEMP = V1:V3 - 2 创建名称分别为 TEMP、VX 和 VY 的三个新列。TEMP 列中的值是 V1 列内容减去 2 所得的结果，VX 列的值是 V2 列内容减去 2 所得的结果，而 VY 列的值是 V3 列内容减去 2 所得的结果。
TEMP = V1 - V1 创建名为 TEMP 的新列，其中包含的值全为 0（因为任何数字减去自身的结果均为 0）。
TEMP = V1 - V2 创建名为 TEMP 的新列，其中每个值都是列 V1 的行值减去列 V2 的相应行值所得的结果。
TEMP = V1:V3 -V4:V6 创建名称分别为 TEMP、VX 和 VY 的三个新列。列 TEMP 包含 V1 中的值减去列 V4 的相应行值所得的结果。列 VX 包含从 V2 中减去列 V5 所得的结果。列 VY 包含从 V3 中减去列 V6 所得的结果。
TEMP = V1[10:20] - V2 或 TEMP = V1[10:20] - V2[1:11] 创建名为 TEMP 的新列，其中，前 11 个单元格包含 V1 列第 10-20 行的值减去 V2 列第 1-11 行的值所得的结果。TEMP 中的其他单元格为空。

## 相关函数

函数	描述
PLUS	将两个数据范围的内容相加
SUM 或 TOTAL	计算某个范围的单元格的总和

---

## MOD 宏

MOD 宏在 IBM Campaign 和 IBM Interact 中都可用。

### 语法

data MOD divisor data % divisor

### 参数

data

要计算模数的整数值。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

divisor

用于计算模数的非零整数基数。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。除非 divisor 是常量，否则 divisor 中的列数必须与 data 中的列数相等。要了解 divisor 的格式定义（与 data 相同），请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

## 描述

MOD 计算所指定数据范围除以指定值的余数。其计算方法是，将每个值除以 divisor，并返回余数。它针对每个输入列返回一个新列，其中每个新列都包含 data 中的数字以 divisor 取模所得的结果。余数的符号与 data 相同（正或负）。

如果 divisor 是常量，那么所指定列中的每个值都会以该值取模。如果 divisor 是一个列，那么将逐行执行计算。data 中的值将以 divisor 的第一行值取模，第二行以第二行取模，依此类推。这种逐行计算会针对每一行生成一个结果，直至处理完最短列中的最后一个值为止。

注：如果 divisor 为零，那么将返回除零错误。

注：MOD 运算符可以简写为百分号 (%)。例如，TEMP = 5 % 3 等同于 TEMP = 5 MOD 3。

## 示例

TEMP = 10 MOD 8 或 TEMP = 10 % 8 创建名为 TEMP 的新列，其中包含值 2。
TEMP = -10 % 8 创建名为 TEMP 的新列，其中包含值 -2。
TEMP = V1 % 8 创建名为 TEMP 的新列，其中每个值都是 V1 列内容以 8 取模所得的结果。
TEMP = V1:V3 % 2 创建名称分别为 TEMP、VX 和 VY 的三个新列。TEMP 列中的值是 V1 列内容以 2 取模所得的结果值，VX 列的值是 V2 列内容以 2 取模所得的结果值，而 VY 列的值是 V3 列内容以 2 取模所得的结果值。
TEMP = V1 % V1 创建名为 TEMP 的新列，其中针对 V1 列中的每个条目包含一个零。这是因为，每个数字以其自身取模的结果均为零。
TEMP = V1 % V2 创建名为 TEMP 的新列，其中每个值都是列 V1 的行值以列 V2 的相应行值取模所得的结果。请注意，如果 V2=V1，那么与上一个示例相同，返回的值均为零。
TEMP = V1:V3 % V4:V6 创建名称分别为 TEMP、VX 和 VY 的三个新列。列 TEMP 包含 V1 中的值以列 V4 的相应行值取模所得的结果。列 VX 包含列 V2 以 V5 取模所得的结果。列 VY 包含列 V3 以 V6 取模所得的结果。

TEMP = V1[10:20] % V2 或 TEMP = V1[10:20] % V2[1:11]

创建名为 TEMP 的新列，其中，前 11 个单元格是 V1 列第 10-20 行的值以 V2 列第 1-11 行的值取模所得的结果值。TEMP 中的其他单元格为空。

## 相关函数

函数	描述
DIV	将一个指定的数据范围除以另一个数据范围
MOD	计算所指定数据范围中内容的模数

---

## MONTHOF 宏

MONTHOF 宏仅在 IBM Campaign 中可用。

### 语法

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

### 参数

date\_string

表示有效日期的文本。

input\_format

下表中的某个关键字，用于指定 date\_string 的日期格式。

### 描述

MONTHOF 以数字形式返回 date\_string 所指定日期的月份。如果未提供 input\_format，那么将使用缺省关键字 DELIM\_M\_D\_Y。

### 示例

MONTHOF("012171",MMDDYY) 返回数字 1。

注：请参阅第 46 页的『DATE』，以获取有效日期格式的更多相关信息。

## 相关函数

函数	描述
DAYOF	以数字形式返回某个日期是一个月中的第几天。
WEEKDAYOF	以数字形式返回某个日期是一周中的第几天。
YEAROF	以数字形式返回年份。

## MULT 宏

MULT 宏在 IBM Campaign 和 IBM Interact 中都可用。

### 语法

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

### 参数

data

要相乘的数字值。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

multiplier

要与所指定列中的所有值相乘的数字。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。除非 multiplier 是常量，否则 multiplier 中的列数必须与 data 中的列数相等。要了解 multiplier 的格式定义（与 data 相同），请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

### 描述

MULT 将两个所指定数据范围中的值相乘。它针对每个输入列返回一个新列，其中每个新列都包含 data 中的数值乘以 multiplier 所得的结果。如果 multiplier 是常量，那么 data 中的每个值都会与该值相乘。如果 multiplier 是一个列，那么将逐行执行计算。data 中的值将与 multiplier 的第一行值相乘，第二行与第二行相乘，依此类推。这种逐行计算会针对每一行生成一个结果，直至处理完最短列中的最后一个值为止。

注：使用每一行中的数字  $x$  都与 multiplier 相同的列等同于使用常量  $x$  作为 multiplier。

注：MULT 运算符可以简写为星号 (\*)。

### 示例

TEMP = 8 MULT 4 或 TEMP = 8 * 4 创建名为 TEMP 的新列，其中包含值 32。
TEMP = V1 * 8 创建名为 TEMP 的新列，其中每个值都是 V1 列内容乘以 8 所得的结果。
TEMP = V1:V3 * 2 创建名称分别为 TEMP、VX 和 VY 的三个新列。TEMP 列中的值是 V1 列内容乘以 2 所得的结果，VX 列的值是 V2 列内容乘以 2 所得的结果，而 VY 列的值是 V3 列内容乘以 2 所得的结果。
TEMP = V1 * V1 创建名为 TEMP 的新列，其中包含列 V1 中每个值的平方。

TEMP = V1 * V2
创建名为 TEMP 的新列，其中每个值都是列 V1 的行值乘以列 V2 的相应行值所得的结果。
TEMP = V1:V3 * V4:V6
创建名称分别为 TEMP、VX 和 VY 的三个新列。列 TEMP 包含 V1 中的值乘以列 V4 的相应行值所得的结果。列 VX 包含列 V2 乘以 V5 所得的结果。列 VY 包含列 V3 乘以 V6 所得的结果。
TEMP = V1[10:20] * V2 或 TEMP = V1[10:20] * V2[1:11]
创建名为 TEMP 的新列，其中，前 11 个单元格包含 V1 列第 10-20 行的值乘以 V2 列第 1-11 行的值所得的结果。TEMP 中的其他单元格为空。

## 相关函数

函数	描述
DIV	将一个指定的数据范围除以另一个数据范围
EXP	以自然数 (e) 为底数，并以所指定数据范围中的每个单元格内容为指数，计算乘幂结果
POW	按指定的指数幂，计算底数值的乘幂结果

## NE 宏

NE 宏在 IBM Campaign 和 IBM Interact 中都可用。

### 语法

data1 NE data2 data1 != data2 data1 <> data2

### 参数

data1

要比较的单元格范围。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

data2

要与所指定列中的所有值进行比较的数字。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。除非 data2 是常量，否则 data2 中的列数必须与 data1 中的列数相等。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

### 描述

NE 对两个指定的数据范围进行比较，如果值不相等则返回 1，否则返回 0。它针对每个输入列返回一个新列，其中每个新列都包含 data1 的相应列与 data2 的相应列进行比较所得的结果（即，data1 的第一列与 data2 的第一列进行比较，第二列与第二列进行比较，依此类推）。

如果 data2 是常量，那么 data1 中的每个值都会与该值进行比较。如果 data2 是一个列，那么将逐行执行计算。data1 的第一行值将与 data2 的第一行值进行比较，第二行与第二行进行比较，依此类推。这种逐行计算会针对每一行生成一个结果，直至处理完最短列中的最后一个值为止。

注：使用每一行中的数字  $x$  都与 data2 相同的列等同于使用常量  $x$  作为 data2。

注：NE 运算符可以简写为惊叹号后跟等号 (!=)，或简写为小于号后跟大于号 (<>)。

## 示例

TEMP = 3 NE 4 或 TEMP = 3 != 4 TEMP = 3 <> 4
创建名为 TEMP 的新列，其中包含值 1（因为 3 不等于 4）。
TEMP = V1 != 8
创建名为 TEMP 的新列，其中每个值都是 1（如果列 V1 的相应行值不等于数字 8）或者 0。
TEMP = V1:V3 != 2
创建名称分别为 TEMP、VX 和 VY 的三个新列。TEMP 列中的值是 V1 列内容与值 2 进行比较的结果，VX 列的值是 V2 列内容与值 2 进行比较的结果，而 VY 列的值是 V3 列内容与值 2 进行比较的结果。
TEMP = V1 != V1
创建名为 TEMP 的新列，其中包含的值全为 0（因为每个数字都与其自身相等）。
TEMP = V1 != V2
创建名为 TEMP 的新列，其中每个值都是列 V1 的行值与列 V2 的相应行值进行比较所得的结果。
TEMP = V1:V3 != V4:V6
创建名称分别为 TEMP、VX 和 VY 的三个新列。列 TEMP 包含 V1 中的值与列 V4 的相应行值进行比较所得的结果。列 VX 包含列 V2 与 V5 进行比较的结果。列 VY 包含列 V3 与 V6 进行比较的结果。
TEMP = V1[10:20] != V2 或 TEMP = V1[10:20] != V2[1:11]
创建名为 TEMP 的新列，其中，前 11 个单元格包含 V1 列第 10-20 行的值与 V2 列第 1-11 行进行比较所得的结果。TEMP 中的其他单元格为空。

## 相关函数

函数	描述
EQ	如果一个数据范围等于另一个数据范围，那么返回 TRUE
GE	如果一个数据范围大于或等于另一个数据范围，那么返回 TRUE
GT	如果一个数据范围大于另一个数据范围，那么返回 TRUE
LE	如果一个数据范围小于或等于另一个数据范围，那么返回 TRUE
LT	如果一个数据范围小于另一个数据范围，那么返回 TRUE

## NOT 宏

NOT 宏在 IBM Campaign 和 IBM Interact 中都可用。

### 语法

NOT(data) ! data

### 参数

data

要计算逻辑 NOT 运算结果的数字值。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

### 描述

NOT 返回所指定数据范围内值的逻辑 NOT 运算结果。它针对每个输入列返回一个新列，其中每个新列都包含相应输入列中值的逻辑 NOT 运算结果。对于非零值，此函数返回 0，对于零值返回 1。

注：NOT 运算符可以简写为惊叹号 (!)。请在数据值之前使用惊叹号（例如，要指定 NOT (V1)，输入 !V1 即可）。

### 示例

TEMP = NOT(3.2) 或 TEMP = !1 创建名为 TEMP 的新列，其中包含值 0。
TEMP = !0 或 TEMP = !(2+2=3) 创建名为 TEMP 的新列，其中包含值 1。
TEMP = !V1 创建名为 TEMP 的新列，其中每个值都是 V1 列中值的逻辑 NOT 运算结果。
TEMP = !V1:V3 创建名称分别为 TEMP、VX 和 VY 的三个新列。TEMP 列中的值是 V1 列中值的逻辑 NOT 运算结果，VX 列的值是 V2 列中值的逻辑 NOT 运算结果，而 VY 列的值是 V3 列中值的逻辑 NOT 运算结果。
TEMP = !V1[10:20] 创建名为 TEMP 的新列，其中，前 11 个单元格包含 V1 列第 10-20 行值的逻辑 NOT 运算结果。TEMP 中的其他单元格为空。
TEMP = !V1[1:5]:V2 创建名称分别为 TEMP 和 VX 的两个新列，每个列在第 1-5 行都有值（其他单元格为空）。列 TEMP 中的值是列 V1 中相应行值的逻辑 NOT 运算结果，而列 VX 中的值是列 V2 中相应行值的逻辑 NOT 运算结果。

## 相关函数

函数	描述
AND	计算两个指定数据范围之间的逻辑 AND 运算结果
INVERSE	计算所指定数据范围中内容的相反数
OR	计算两个指定数据范围之间的逻辑 OR 运算结果
SIGN	计算所指定数据范围内值的符号（正或负）

---

## NUMBER 宏

NUMBER 宏在 IBM Campaign 和 IBM Interact 中可用。

### 语法

NUMBER(data [, conversion\_keyword])

### 参数

data

要转换为数字值的 ASCII 文本数据。此参数可以用引号括起来的 ASCII 文本、文本列、包含文本的单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

conversion\_keyword

这个可选关键字指定如何解释日期和时间的文本格式。请选择下表中的某个关键字。

注：如果未指定此参数，那么缺省情况下将使用 1。

转换关键字	格式	描述
0	#####	将每个文本字符串的前五个字符转换为唯一的数字
1	\$ (缺省值)	将美元值转换为数字（例如，将 "\$123.45" 转换为 123.45）
2	%	将百分比值转换为数字（例如，将 "50%" 转换为 0.5）
3	mm/dd/yy hh:mm	将日期时间转换为 0000 年 1 月 1 日以来经过的天数（yy 年份会自动加上 1900）
4	dd-mmm-yy	将日期转换为 0000 年 1 月 1 日以来经过的天数（yy 年份会自动加上 1900）
5	mm/dd/yy	将日期转换为 0000 年 1 月 1 日以来经过的天数（yy 年份会自动加上 1900）
6	mmm-yy	将日期转换为所指定月份第一天与 0000 年 1 月 1 日之间经过的天数（yy 年份会自动加上 1900）

转换关键字	格式	描述
7	dd-mmm	将日期转换为本年以来经过的天数（例如，将 "01-FEB" 转换为 32）
8	mmm	将三字母月份缩写转换为 1-12 之间的值（例如，将 "DEC" 转换为 12）
9	{January   February   March ... }	将完整拼写的月份名称转换为 1-12 之间的值（例如，将 "March" 转换为 3）
10	{Sun   Mon   Tue ... }	将三字母工作日缩写转换为 0-6 之间的值，其中每周以星期日开始（例如，将 "Sun" 转换为 0）
11	{Sunday   Monday   Tuesday ... }	将完整拼写的工作日名称转换为 0-6 之间的值，其中每周以星期日开始（例如，将 "Monday" 转换为 1）
12	hh:mm:ss {AM   PM}	将时间转换为上午 00:00:00（午夜）以来经过的秒数（例如，将 "01:00:00 AM" 转换为 3600）
13	hh:mm:ss	将时间转换为上午 00:00:00（午夜）以来经过的秒数（例如，将 "01:00:00" 转换为 3600）
14	hh:mm {AM   PM}	将时间转换为上午 00:00:00（午夜）以来经过的分钟数（例如，将 "01:00 AM" 转换为 60）
15	hh:mm	将时间转换为上午 00:00:00（午夜）以来经过的分钟数（例如，将 "01:00" 转换为 60）
16	mm:ss	将时间转换为上午 00:00:00（午夜）以来经过的秒数（例如，将 "30:00" 转换为 1800）
17	ddmm	将日期转换为本年以来经过的天数（例如，将 "3101" 转换为 31）
18	ddmmm	将日期转换为本年以来经过的天数（例如，将 "31JAN" 转换为 31）
19	ddmmyy	将日期转换为 0000 年 1 月 1 日以来经过的天数（如果 yy 小于或等于 20，那么年份将自动加上 1900，否则加上 2000）
20	ddmmyyyy	将日期转换为 0000 年 1 月 1 日以来经过的天数（例如，将 "31JAN0000" 转换为 31）
21	ddmmyy	将日期转换为 0000 年 1 月 1 日以来经过的天数（如果 yy 小于或等于 20，那么年份将自动加上 1900，否则加上 2000）
22	ddmmyyyy	将日期转换为 0000 年 1 月 1 日以来经过的天数（例如，将 "31010000" 转换为 31）
23	mmdd	将日期转换为本年以来经过的天数（例如，将 "0131" 转换为 31）
24	mmddy	将日期转换为 0000 年 1 月 1 日以来经过的天数（如果 yy 小于或等于 20，那么年份将自动加上 1900，否则加上 2000）
25	mmddyyyy	将日期转换为 0000 年 1 月 1 日以来经过的天数（例如，将 "01010001" 转换为 366）

转换关键字	格式	描述
26	mmm	将三字母月份缩写转换为 1-12 之间的值 (例如, 将 "MAR" 转换为 3) [请注意, 这等同于转换关键字 8]
27	mmdd	将日期转换为今年以来经过的天数 (例如, 将 "JAN31" 转换为 31)
28	mmddyy	将日期转换为 0000 年 1 月 1 日以来经过的天数 (如果 yy 小于或等于 20, 那么年份将自动加上 1900, 否则加上 2000)
29	mmddyyyy	将日期转换为 0000 年 1 月 1 日以来经过的天数 (例如, 将 "FEB010001" 转换为 32)
30	mmmyy	将日期转换为所指定月份第一天与 0000 年 1 月 1 日之间经过的天数 (如果 yy 小于或等于 20, 那么年份将自动加上 1900, 否则加上 2000)
31	mmyyyy	将日期转换为所指定月份第一天与 0000 年 1 月 1 日之间经过的天数 (例如, 将 "FEB0001" 转换为 32)
32	mmyy	将日期转换为所指定月份第一天与 0000 年 1 月 1 日之间经过的天数 (如果 yy 小于或等于 20, 那么年份将自动加上 1900, 否则加上 2000)
33	myyyyy	将日期转换为所指定月份第一天与 0000 年 1 月 1 日之间经过的天数 (例如, 将 "020001" 转换为 32)
34	yymm	将日期转换为所指定月份第一天与 0000 年 1 月 1 日之间经过的天数 (如果 yy 小于或等于 20, 那么年份将自动加上 1900, 否则加上 2000)
35	yymdd	将日期转换为 0000 年 1 月 1 日以来经过的天数 (如果 yy 小于或等于 20, 那么年份将自动加上 1900, 否则加上 2000)
36	yymmm	将日期转换为所指定月份第一天与 0000 年 1 月 1 日之间经过的天数 (如果 yy 小于或等于 20, 那么年份将自动加上 1900, 否则加上 2000)
37	yymddd	将日期转换为 0000 年 1 月 1 日以来经过的天数 (如果 yy 小于或等于 20, 那么年份将自动加上 1900, 否则加上 2000)
38	yyyy	将年份转换为 0000 年以来经过的年数 (例如, 将 "1998" 转换为 1998)
39	yyyymm	将日期转换为所指定月份第一天与 0000 年 1 月 1 日之间经过的天数 (例如, 将 "000102" 转换为 32)
40	yyyymddd	将日期转换为 0000 年 1 月 1 日以来经过的天数 (例如, 将 "00010201" 转换为 32)

转换关键字	格式	描述
41	yyyymm	将日期转换为所指定月份第一天与 0000 年 1 月 1 日之间经过的天数（例如，将 "000102" 转换为 32）
42	yyyymmdd	将日期转换为 0000 年 1 月 1 日以来经过的天数（例如，将 "0001FEB01" 转换为 32）
43	<day>* <month>	将依次包含日子和月份的任何定界日期转换为本年以来经过的天数（例如，将 "15-JAN" 转换为 15）
44	<day>* <month>* <year>	将依次包含日子、月份和年份的任何定界日期转换为 0000 年 1 月 1 日以来经过的天数（例如，将 "1/1/0001" 转换为 366）
45	<month>* <day>	将依次包含月份和日子的任何定界日期转换为本年以来经过的天数（例如，将 "JAN 31" 转换为 31）
46	<month>* <day>* <year>	将依次包含月份、日子和年份的任何定界日期转换为 0000 年 1 月 1 日以来经过的天数（例如，将 "JAN 1, 0001" 转换为 366）
47	<month>* <year>	将依次包含月份和年份的任何定界日期转换为所指定月份第一天与 0000 年 1 月 1 日之间经过的天数
48	<year>* <month>	将依次包含年份和月份的任何定界日期转换为所指定月份第一天与 0000 年 1 月 1 日之间经过的天数
49	<year>* <month>* <day>	将依次包含年份、月份和日子的任何定界日期转换为 0000 年 1 月 1 日以来经过的天数（例如，将 "0001/01/01" 转换为 366）
50	yy	将年份转换为 0000 年以来经过的年数（例如，将 "97" 转换为 97）
51	mm	将月份转换为 1-12 之间的值（例如，将 "SEP" 转换为 9）
52	dd	将日子转换为 1-31 之间的值（例如，将 "28" 转换为 28）
53	{January   February   March ... }	将完整拼写的月份名称转换为 1-12 之间的值（例如，将 "March" 转换为 3）[请注意，这等同于转换关键字 9]
54	{Sunday   Monday   Tuesday ... }	将完整拼写的工作日名称转换为 1-7 之间的值，其中每周以星期日开始（例如，将 "Sunday" 转换为 1）
55	{Sun   Mon   Tue ... }	将三字母工作日缩写转换为 1-7 之间的值，其中每周以星期日开始（例如，将 "Sun" 转换为 1）

## 描述

NUMBER 通过使用指定的格式来转换日期和时间，将所指定数据范围内的文本值转换为数值。如果无法使用指定的 conversion\_keyword 解析文本字符串，那么 NUMBER 将生

成错误。对于每个唯一的文本字符串，格式 0 会将每个文本字符串的前五个字符转换为不同数字。这是将一系列文本更改为唯一类，以便输出到分类器的一种简单方法。

定界格式（转换关键字 43-49）支持下列全部定界符：

- /（斜杠）
- -（短划线）
- ,（逗号）
- " "（空格）
- :（冒号）

月份可以表示为 mm 或 mmm；日子可以表示为 d 或 dd；年份可以表示为 yy 或 yyyy。

注：为了支持 2000 年一致性，日期中的所有年份都可指定为 yyyy 以替代 yy。为了实现向后兼容，在转换关键字 1-16 中，yy（2 位数年份）会自动加上 1900。对于转换关键字 17-55，小于 threshold 的 yy 会自动加上 2000；而大于或等于 threshold 的 yy 会自动加上 1900。

注：2000 年阈值 threshold 可在高级设置窗口（通过选项 > 设置 > 高级设置来调用）的数据清除选项卡中设置。

注：更改 2000 年阈值后，必须使用 NUMBER 宏函数更新所有宏函数，以便处理含两位数年份的日期值。要强制更新宏函数，您可以进行任何编辑（例如，添加空格并将其删除），然后单击勾选标记图标以接受更改。

注：使用格式 0 时，将仅使用每个文本字符串的前五个字符来生成唯一数字。前五个字符相同的所有字符串将转换为同一个数字值。同一个文本字符串每次都会生成相同的数字值，即使在不同的电子表格上使用也一样。如有需要，请使用字符串宏来处理字符串，以使前五个字符唯一地定义类。请注意，所生成的数字值可能非常小。使用显示格式窗口可增加所显示的小数位数，或将格式更改为指数方式 (00E+00)。

## 示例

TEMP = NUMBER("\$1.23") 或 TEMP = NUMBER("123%", 2) 创建名为 TEMP 的新列，其中包含数字 1.23。
TEMP = NUMBER(column("Jan", "Mar", "Dec", 8)) 创建名为 TEMP 的新列，其中包含数字 1、3 和 12。
TEMP = NUMBER("1:52 PM", 14) 创建名为 TEMP 的新列，其中包含数字 832。
TEMP = NUMBER("1/1/95", 5) 创建名为 TEMP 的新列，其中包含数字 728660。
TEMP = NUMBER(V1) 创建名为 TEMP 的新列，其中包含列 V1 的文本字符串的数字值。任何美元值都将正确转换为数字值。针对无法使用 \$ 格式解析的文本字符串，将会返回 ???。

TEMP = NUMBER(V1:V3, 4)

创建名称分别为 TEMP、VX 和 VY 的三个新列。列 TEMP 包含列 V1 的文本字符串的数字值。列 VX 包含列 V2 的文本字符串的数字值。列 VY 包含列 V3 的文本字符串的数字值。dd-mmm-yy 格式的任何日期都将转换为相对于 0000 年 1 月 1 日的天数偏移量。针对无法使用 \$ 格式解析的文本字符串，将会返回 ???。

TEMP = NUMBER(V1[10:20]:V2, 10)

创建名称分别为 TEMP 和 VX 的两个新列。列 TEMP 包含 V1 列第 10-20 行的文本字符串的数字值。列 VX 包含 V2 列第 10-20 行的文本字符串的数字值。星期几的所有标准三字符表示都将转换为数字 0-6 (0 表示星期日, 6 表示星期六)。没有工作日名称的匹配项时, 将返回 ???。

TEMP = NUMBER(V1, 0)

假定列 V1 中的值全都是 5 位的文本字符串, 在这种情况下, 将会创建名为 TEMP 的一个新列, 其中会针对每个唯一字符串包含一个不同的数字值。

## 相关函数

函数	描述
WEEKDAY	将 ASCII 文本日期字符串转换为一周中的第几天

---

## OR 宏

OR 宏在 IBM Campaign 和 IBM Interact 中都可用。

### 语法

data1 OR data2 data1 || data2

### 参数

data1

要与 data2 中的值进行逻辑 OR 运算的数字。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义, 请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

data2

要与 data1 中的值进行逻辑 OR 运算的数字。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。除非 data2 是常量, 否则 data2 中的列数必须与 data1 中的列数相等。要了解 data 的格式定义, 请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

### 描述

OR 计算两个所指定数据范围之间的逻辑 OR 运算结果。它针对每个输入列返回一个新列, 其中每个新列都包含 data1 的相应列与 data2 的相应列进行逻辑 OR 运算的结果 (即, data1 的第一列与 data2 的第一列进行逻辑 OR 运算, 第二列与第二列进行逻辑 OR 运算, 依此类推)。

如果 data2 是常量，那么 data1 中的每个值都与该值进行逻辑 OR 运算。如果 data2 包含一个或多个列，那么将在 data1 中的一列与 data2 中的一列之间逐行执行计算。data1 的第一行与 data2 的第一个行值进行逻辑 OR 运算，第二行与第二行进行逻辑 OR 运算，依此类推。这种逐行计算会针对每一行生成一个结果，直至处理完最短列中的最后一个值为止。

注：使用每一行中的数字  $x$  都与 data2 相同的列等同于使用常量  $x$  作为 data2。

注：OR 运算符可以简写为双竖线 (||)。请使用双竖线分隔两个自变量（例如，要指定 V1 OR 3，输入 V1||3 即可）。

## 示例

TEMP = 1 OR 8 或 TEMP = 1    8
创建名为 TEMP 的新列，其中包含值 1（任何非零数字都视为 1）。
TEMP = V1    1
创建名为 TEMP 的新列，其中包含的值全为 1（每个值与数字 1 执行 OR 运算会生成 1）。
TEMP = V1    V2
创建名为 TEMP 的新列，其中每个值都是列 V1 的行值与列 V2 的相应行值进行逻辑 OR 运算的结果。
TEMP = V1:V3    V4:V6
创建名称分别为 TEMP、VX 和 VY 的三个新列。列 TEMP 包含 V1 中的值与列 V4 的相应行值进行逻辑 OR 运算的结果。列 VX 包含列 V2 与 V5 进行逻辑 OR 运算所生成的值。列 VY 包含列 V3 与 V6 进行逻辑 OR 运算所生成的值。
TEMP = V1[10:20]    V2
创建名为 TEMP 的新列，其中，前 11 个单元格包含 V1 列第 10-20 行值与 V2 列执行逻辑 OR 运算的结果。TEMP 中的其他单元格为空。

## 相关函数

函数	描述
AND	计算两个指定数据范围之间的逻辑 AND 运算结果
NOT	计算所指定数据范围中内容的逻辑 NOT 运算结果

## POSITION 宏

POSITION 宏仅在 IBM Campaign 中可用。

### 语法

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

## 参数

colName

列的值（必须是 string 类型）。

pattern

要搜索的模式或字符串。

start

搜索开始字节。

occurrence

请为 n 指定值，以搜索该模式的第 n 次出现并返回其位置。

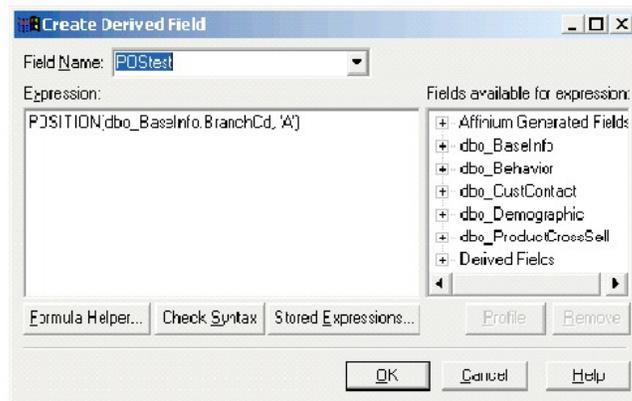
## 描述

POSITION 返回模式或字符串在列 (colName) 的值（必须为 string 类型）中的起始字节位置。如果指定了 start，那么将从该处开始搜索。Occurrence 是要返回的模式的第 n 次出现。

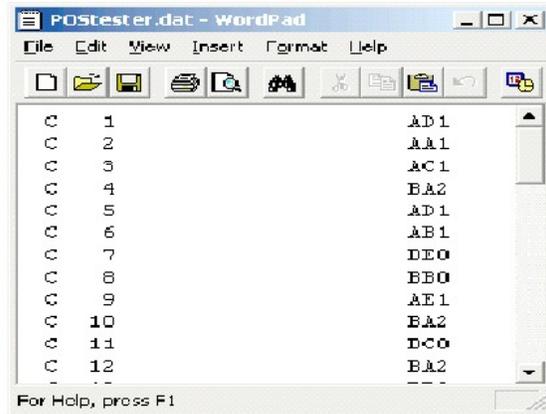
注：该搜索不区分大小写。

## 示例

在以下示例中，我们要在列 dbo\_BaseInfo.BranchCd 的值中搜索模式或字符串"A"，并将返回的值赋予派生字段 P0Stest。



以下示例显示了表中的一些行，其中 dbo\_BaseInfo.BranchCd 和 P0Stest 的值并排显示。



下面是一个更复杂的示例：

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

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

这将返回在"X"第二次出现之后，CellCode 的值以"AAA"结尾的行。

---

## PLUS 宏

PLUS 宏在 IBM Campaign 和 IBM Interact 中都可用。

### 语法

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

### 参数

data

包含被加数的单元格范围。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

addend

要与所指定列中的所有值相加的数值。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。除非 data2 是常量，否则 data2 中的列数必须与 data1 中的列数相等。要了解 addend 的格式定义（与 data 相同），请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

### 描述

PLUS 将两个所指定数据范围中的值相加。它针对每个输入列返回一个新列，其中每个新列都包含 data1 的相应列与 data2 的相应列相加所得的结果（即，data1 的第一列与 data2 的第一列相加，第二列与第二列相加，依此类推）。

如果 data2 是常量，那么 data1 中的每个值都会加上该值。如果 data2 包含一个或多个列，那么将在 data1 中的一列与 data2 中的一列之间逐行执行计算。data1 的第一行与 data2 的第一个行值相加，第二行与第二行相加，依此类推。这种逐行计算会针对每一行生成一个结果，直至处理完最短列中的最后一个值为止。

注：PLUS 运算符可以简写为加号 (+)。

## 示例

TEMP = 3 PLUS 4 或 TEMP = 3 + 4
创建名为 TEMP 的新列，其中包含值 7。
TEMP = V1 + 8
创建名为 TEMP 的新列，其中每个值都是 V1 列内容加上 8 所得的结果。
TEMP = V1 + V1
创建名为 TEMP 的新列，其中包含 V1 列内容的两倍值。
TEMP = V1 + V2
创建名为 TEMP 的新列，其中每个值都是列 V1 的行值加上列 V2 的相应行值所得的结果。
TEMP = V1:V3 + V4:V6
创建名称分别为 TEMP、VX 和 VY 的三个新列。列 TEMP 包含 V1 中的值加上列 V4 的相应行值所得的结果。列 VX 包含列 V2 加上列 V5 所得的结果。列 VY 包含列 V5 加上列 V6 所得的结果。
TEMP = V1[10:20] + V2 或 TEMP = V1[10:20] + V2[1:11]
创建名为 TEMP 的新列，其中，前 11 个单元格包含 v1 列第 10-20 行的值加上 V2 列第 1-11 行的值所得的结果。TEMP 中的其他单元格为空。

## 相关函数

函数	描述
MINUS	从一个指定的数据范围中减去另一个数据范围
SUM 或 TOTAL	计算某个范围的单元格的总和

## POW 宏

POW 宏在 IBM Campaign 和 IBM Interact 中都可用。

### 语法

base POW exponent base ^ exponent

### 参数

base

要计算乘幂结果的数字值。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。要了解 base 的格式定义（与 data 相同），请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

exponent

用于对 data 中的值执行乘幂运算的指数。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。exponent 中的列数必须等于 base 中的列数，否则 base 是常量。要了解 exponent 的格式定义（与 data 相同），请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

## 描述

POW 以第一个数据范围中的值为底数，以第二个数据范围中指定的幂计算乘幂结果（即，计算  $base^{exponent}$ ）。它针对每个输入列返回一个新列，其中每个新列都包含 base 以 exponent 为指数计算而得的乘幂结果（即，data1 的第一列以 data 的第一列为指数，第二列以第二列为指数，依此类推）。

如果 exponent 是常量，那么 base 中的每个值以该值为指数。如果 exponent 包含一个或多个列，那么将在 base 中的一列与 exponent 中的一列之间逐行执行计算。base 的第一行以 exponent 的第一行值为指数，第二行以第二行为指数，依此类推。这种逐行计算会针对每一行生成一个结果，直至处理完最短列中的最后一个值为止。

注：POW 运算符可以简写为音调符号 (^)。例如，TEMP = 2^8 等同于 TEMP = 2 POW 8。

注：当值  $x$  过大或过小时，将会返回溢出。base^exponent 超过最大或最小 32 位浮点值时就会发生这种情况。

## 示例

TEMP = 2 POW 3 或 TEMP = 2^3 创建名为 TEMP 的新列，其中包含值 8。
TEMP = V1 ^ 0.5 创建名为 TEMP 的新列，其中每个值都是 V1 列内容的平方根（这等同于 SQRT(V1)）。
TEMP = V1 ^ V3 创建名为 TEMP 的新列，其中每个值都是列 V1 的行值以列 V2 的相应行值为指数的乘幂结果。
TEMP = V1:V3 ^ V4:V6 创建名称分别为 TEMP、VX 和 VY 的三个新列。列 TEMP 包含 V1 中的值以列 V4 的相应行值为指数的乘幂结果。列 VX 包含列 V2 以列 V5 中的相应值为指数的乘幂结果。列 VY 包含列 V3 以 V6 的相应值为指数的乘幂结果。
TEMP = V1[10:20] POW V2 或 TEMP = V1[10:20] POW V2[1:11] 创建名为 TEMP 的新列，其中，前 11 个单元格包含 V1 列第 10-20 行的值以列 V2 中第 1-11 行的值为指数的乘幂结果。TEMP 中的其他单元格为空。

## 相关函数

函数	描述
EXP	以自然数 (e) 为底数，并以所指定数据范围中的每个单元格内容为指数，计算乘幂结果
LN 或 LOG	计算所指定数据范围中内容的自然对数
LN2	计算所指定数据范围中内容以 2 为底的对数
LN10	计算所指定数据范围中内容以 10 为底的对数

---

## RANDOM 宏

RANDOM 宏仅在 IBM Campaign 中可用。

### 语法

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

### 参数

num

要生成的随机数的数目。此值必须是大于零的正整数。

value1

要生成的随机数的一个边界。此参数可以是任何常量值，也可以是求值为常量的表达式。如果未提供此参数，那么缺省值为零。

value2

要生成的随机数的另一个边界。此参数可以是任何常量值，也可以是求值为常量的表达式。如果未提供此参数，那么缺省值为 1。

seed

用于生成随机数的可选种子值。此值必须是整数。

### 描述

RANDOM 将生成一个随机数列。它返回一个新列，其中包含 num 个随机数。如果指定了 value1 和 value2，那么将生成介于这两个边界（含边界本身）之间的随机数。如果未指定这两个参数，那么缺省情况下将生成介于 0 与 1 之间的值。如果提供了 seed，那么它将用作随机数字生成器的种子值。

注：如果 seed 大于或等于  $2^{32}$ ，那么将用  $2^{32} - 1$  代替该值。大于  $2^{24}$  的 seed 值将进行舍入（即，去掉精度）。因此，多个值可能会生成同一个 seed 值。

## 示例

TEMP = RANDOM() 创建名为 TEMP 的新列，其中包含长度不受限制的随机数。
TEMP = RANDOM(100) 创建名为 TEMP 的新列，其中包含 100 个介于 0.0 与 1.0 之间的随机数。
TEMP = RANDOM(100, 5943049) 创建名为 TEMP 的新列，其中包含 100 个根据种子数字 5943049 生成的随机数。
TEMP = RANDOM(100, 0, 100) 创建名为 TEMP 的新列，其中包含 100 个介于 0 与 100.0 之间的随机数。
TEMP = RANDOM(100, 0, 100, 5943049) 创建名为 TEMP 的新列，其中包含 100 个根据种子数字 5943049 生成且介于 -0 与 100 之间的随机数。

## 相关函数

函数	描述
RANDOM_GAUSS	根据高斯分布返回指定数目的随机值

---

## RANDOM\_GAUSS 宏

RANDOM\_GAUSS 宏仅在 IBM Campaign 中可用。

### 语法

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

### 参数

num

要生成的随机数的数目。此值必须是大于零的正整数。

mean

高斯分布的平均值。此参数可以是任何常量值，也可以是求值为常量的表达式。如果未提供此参数，那么缺省值为零。

std

高斯分布的标准差。此参数可以是任何常量值，也可以是求值为常量的表达式。如果未提供此参数，那么缺省值为 1。

seed

用于生成随机数的可选种子值。此值必须是整数。（如果提供了非整数值，那么会改为自动使用该值的向下取整。）

## 描述

RANDOM\_GAUSS 将根据高斯分布生成一个随机数列。它返回一个新列，其中包含 num 个随机数。如果指定了 mean 和 std，那么将使用具有所指定的平均值和标准差的高斯分布来生成随机数。如果未指定这两个参数，那么缺省高斯分布的平均值为 0，标准差为 1。如果提供了 seed，那么它将用作随机数字生成器的种子值。

## 示例

TEMP = RANDOM_GAUSS(100)
创建名为 TEMP 的新列，其中包含 100 个值，它们是从平均值为 0 的单位标准差高斯分布中随机取样而得。
TEMP = RANDOM_GAUSS(500, 3)
创建名为 TEMP 的新列，其中包含 100 个值，它们是从平均值为 0 的单位标准差高斯分布中随机取样而得。数字 3 用作随机数字生成器的种子值。
TEMP = RANDOM_GAUSS(5000, 100, 32)
创建名为 TEMP 的新列，其中包含 5000 个值，它们是从平均值为 100 且标准差为 32 的高斯分布中随机取样而得。
TEMP = RANDOM_GAUSS(500, -1, 2, 3)
创建名为 TEMP 的新列，其中包含 500 个值，它们是从平均值为 -1 且标准差为 2 的高斯分布中随机取样而得。数字 3 用作随机数字生成器的种子值。

## 相关函数

函数	描述
RANDOM	返回指定数目的随机数

---

## ROUND 宏

ROUND 宏仅在 IBM Campaign 中可用。

### 语法

ROUND(data)

### 参数

data

要舍入的数字值。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

### 描述

ROUND 将所指定数据范围内的值舍入到最接近的整数。它针对每个输入列返回一个新列，其中每个新列都包含相应输入列中数字的取整值。正好位于中间的数字将向上舍入（例如，2.5 舍入成 3.0，而 -2.5 舍入成 -2.0）。

## 示例

TEMP = ROUND(3.2)
创建名为 TEMP 的新列，其中包含值 3。
TEMP = ROUND(V1)
创建名为 TEMP 的新列，其中每个值都是 V1 列内容的取整值。
TEMP = ROUND(V1:V3)
创建名称分别为 TEMP、VX 和 VY 的三个新列。TEMP 列中的值是 V1 列内容的取整值，VX 列的值是 V2 列内容的取整值，而 VY 列的值是 V3 列内容的取整值。
TEMP = ROUND(V1[10:20])
创建名为 TEMP 的新列，其中，前 11 个单元格包含 V1 列第 10-20 行中的取整值。TEMP 中的其他单元格为空。
TEMP = ROUND(V1[1:5]:V2)
创建名称分别为 TEMP 和 VX 的两个新列，每个列在第 1-5 行都有值（其他单元格为空）。列 TEMP 中的值是列 V1 中相应行的取整值，而列 VX 中的值是列 V2 中相应行的取整值。

## 相关函数

函数	描述
INT	计算所指定数据范围中内容的整数值（向下舍入）
MOD	计算所指定数据范围中内容的模数
TRUNCATE	返回所指定数据范围内每个值的非小数部分

---

## ROWNUM 宏

ROWNUM 宏仅在 IBM Campaign 中可用。

### 语法

ROWNUM()

### 描述

ROWNUM 生成从 1 到记录数的序号。第一条记录的编号为 1，第二条记录的编号为 2，依此类推。

注：ROWNUM 可处理的最大记录数为 20 亿。

---

## RTRIM 宏

RTRIM 宏在 IBM Campaign 和 IBM Interact 中可用。

### 语法

RTRIM(data)

## 参数

data

## 描述

RTRIM 从所指定数据范围的每个字符串值中除去尾部空格字符，并返回转换后的字符串。它针对每个输入列返回一个新列。

## 示例

```
Temp = RTRIM "gold "
```

创建名为 Temp 的新字符串，其中包含"gold"。

---

## SIGN 宏

SIGN 宏仅在 IBM Campaign 中可用。

## 语法

SIGN(data)

## 参数

data

要计算符号的数字值。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

## 描述

SIGN 检验所指定数据范围内值的符号。它针对每个输入列返回一个新列，其中每个新列都包含相应输入列中数字的符号。对于所有大于零的值，将返回 +1；对于所有小于零的值，将返回 -1；对于值零，将返回零。

## 示例

```
TEMP = SIGN(-3)
```

创建名为 TEMP 的新列，其中包含值 -1。

```
TEMP = SIGN(V1)
```

创建名为 TEMP 的新列，其中每个值都是 V1 列内容的符号。

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

创建名称分别为 TEMP、VX 和 VY 的三个新列。TEMP 列中的值是 V1 列内容的符号，VX 列的值是 V2 列内容的符号，而 VY 列的值是 V3 列内容的符号。

```
TEMP = SIGN(V1[10:20])
```

创建名为 TEMP 的新列，其中，前 11 个单元格包含 V1 列第 10-20 行值的符号。TEMP 中的其他单元格为空。

```
TEMP = SIGN(V1[10:50]:V2)
```

创建名称分别为 TEMP 和 VX 的两个新列，每个列在第 1-41 行中都有值（其他单元格为空）。列 TEMP 中的值是 V1 列第 10-50 行值的符号，而 VX 列中的值是 V2 列第 10-50 行值的符号。

## 相关函数

函数	描述
ABS	计算所指定数据范围中内容的绝对值
INVERSE	计算所指定数据范围中内容的相反数

---

## SIN 宏

SIN 宏仅在 IBM Campaign 中可用。

### 语法

```
SIN(data [, units_keyword])
```

### 参数

data

要计算正弦值的数字值。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

units\_keyword

这个可选关键字确定输入值和结果是解释为度数还是弧度数。请选择下列其中一项：

RADIAN - 以弧度为单位执行计算（缺省情况）

DEGREE - 以度为单位执行计算

如果未指定此参数，那么缺省情况下将以弧度为单位。（要从弧度转换为度，请除以 PI 并乘以 180。）

有关在 IBM Campaign 中使用关键字的更多详细信息，请参阅第 6 页的『格式规范』。

### 描述

SIN 计算所指定数据范围内值的正弦值。它针对每个输入列返回一个新列，其中每个新列都包含相应输入列中数字的正弦值。

### 示例

```
TEMP = SIN(PI/2)、TEMP = SIN(PI/2, 0) 或 TEMP = SIGN(PI/2, RADIAN)
```

创建名为 TEMP 的新列，其中包含值 1。

TEMP = SIN(V1)
创建名为 TEMP 的新列，其中每个值都是 V1 列内容的正弦值（以弧度计）。
TEMP = SIN(V1:V3, 1) 或 TEMP = SIN(V1:V3, DEGREE)
创建名称分别为 TEMP、VX 和 VY 的三个新列。TEMP 列中的值是 V1 列内容的正弦值，VX 列的值是 V2 列内容的正弦值，而 VY 列的值是 V3 列内容的正弦值。所有值都以度为单位。
TEMP = SIN(V1[10:50]:V2)
创建名称分别为 TEMP 和 VX 的两个新列，每个列在第 1-41 行中都有值（其他单元格为空）。列 TEMP 中的值是 V1 列第 10-50 行值的正弦值，而 VX 列中的值是 V2 列第 10-50 行值的正弦值。所有值都以弧度为单位。

## 相关函数

函数	描述
ASIN	计算所指定数据范围中内容的反正弦值
COS	计算所指定数据范围中内容的余弦值
SINH	计算所指定数据范围中内容的双曲正弦值
TAN	计算所指定数据范围中内容的正切值

---

## SINH 宏

SINH 宏仅在 IBM Campaign 中可用。

### 语法

SINH(data [, units\_keyword])

### 参数

data

要计算双曲正弦值的数字值。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

units\_keyword

这个可选关键字确定输入值和结果是解释为度数还是弧度数。请选择下列其中一项：

RADIAN - 以弧度为单位执行计算（缺省情况）

DEGREE - 以度为单位执行计算

如果未指定此参数，那么缺省情况下将以弧度为单位。（要从弧度转换为度，请除以 PI 并乘以 180。）

有关在 IBM Campaign 中使用关键字的更多详细信息，请参阅第 6 页的『格式规范』。

## 描述

SINH 计算所指定数据范围内值的双曲正弦值。它针对每个输入列返回一个新列，其中每个新列都包含相应输入列中数字的双曲正弦值。对于以弧度为单位的  $x$ ，数字的双曲正弦值为：

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

其中  $e$  是自然数 2.7182818。

注：当值  $x$  过大时，将会返回溢出错误。 $\sinh(x)$  超过最大 32 位浮点值时就会发生这种情况。

## 示例

TEMP = SINH(1)、TEMP = SINH(1, 0) 或 TEMP = SINH(1, RADIAN) 创建名为 TEMP 的新列，其中包含值 1.18。
TEMP = SINH(V1) 创建名为 TEMP 的新列，其中每个值都是 V1 列内容的双曲正弦值（以弧度计）。
TEMP = SINH(V1:V3, 1) 或 TEMP = SINH(V1:V3, DEGREE) 创建名称分别为 TEMP、VX 和 VY 的三个新列。TEMP 列中的值是 V1 列内容的双曲正弦值，VX 列的值是 V2 列内容的双曲正弦值，而 VY 列的值是 V3 列内容的双曲正弦值。所有值都以度为单位。
TEMP = SINH(V1[10:50]:V2) 创建名称分别为 TEMP 和 VX 的两个新列，每个列在第 1-41 行中都有值（其他单元格为空）。列 TEMP 中的值是 V1 列第 10-50 行值的双曲正弦值，而 VX 列中的值是 V2 列第 10-50 行值的双曲正弦值。所有值都以弧度为单位。

## 相关函数

函数	描述
COSH	计算所指定数据范围中内容的双曲余弦值
SIN	计算所指定数据范围中内容的正弦值
TANH	计算所指定数据范围中内容的双曲正切值

---

## SQRT 宏

SQRT 宏仅在 IBM Campaign 中可用。

## 语法

SQRT(data)

### 参数

data

要计算平方根的数字值。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

## 描述

SQRT 计算所指定数据范围内值的平方根。它针对每个输入列返回一个新列，其中每个新列都包含相应输入列中数字的正平方根。

注：如果所定义数据范围内的值是负数，那么将对该单元格返回 ???。

## 示例

TEMP = SQRT(2)
创建名为 TEMP 的新列，其中包含值 1.41。
TEMP = SQRT(V1)
创建名为 TEMP 的新列，其中每个值都是 V1 列内容的平方根。
TEMP = SQRT(V1:V3)
创建名称分别为 TEMP、VX 和 VY 的三个新列。TEMP 列中的值是 V1 列内容的平方根，VX 列的值是 V2 列内容的平方根，而 VY 列的值是 V3 列内容的平方根。
TEMP = SQRT(V1[10:20])
创建名为 TEMP 的新列，其中，前 11 个单元格包含 V1 列第 10-20 行值的平方根。TEMP 中的其他单元格为空。
TEMP = SQRT(V1[10:50]:V2)
创建名称分别为 TEMP 和 VX 的两个新列，每个列在第 1-41 行中都有值（其他单元格为空）。列 TEMP 中的值是 V1 列第 10-50 行值的平方根，而 VX 列中的值是 V2 列第 10-50 行值的平方根。

## 相关函数

函数	描述
DIV	将一个指定的数据范围除以另一个数据范围
MULT	将两个数据范围的内容相乘
POW	按指定的指数幂，计算底数值的乘幂结果

---

## STDV 或 STDEV 宏

STDV 或 STDEV 宏在 IBM Campaign 和 IBM Interact 中都可用。

## 语法

STDV(data [, keyword]) STDEV(data [, keyword])

## 参数

data

要计算标准差的数字值。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

keyword

这个可选关键字确定如何对输入数据范围执行计算。请选择下列其中一项：

ALL - 对 data 中的所有单元格执行计算（缺省值）

COL - 对 data 的每一列单独执行计算

ROW - 对 data 的每一行单独执行计算

有关在 IBM Campaign 中使用关键字的更多详细信息，请参阅第 6 页的『格式规范』。

注：许多宏函数使用关键字参数 {ALL | COL | ROW}。这些关键字在 **IBM Campaign** 中不适用，因为输入数据始终为单个列或字段。这个宏的行为始终像是已指定 COL 关键字一样。因此，使用 **IBM Campaign** 时，无需指定这些关键字。

## 描述

STDV 计算所指定数据范围内所有单元格的标准差。分布的标准差是方差的平方根。标准差的计算方式如下：

$$\sqrt{\frac{1}{n-1} \sum_{j=1}^n (x_j - \text{mean})^2}$$

其中， $x$  是样本， $n$  是样本数，而  $\text{mean}$  是分布的平均值。

注：如果样本数  $n = 1$ ，那么 STDV 将返回错误。

## 示例

```
TEMP = STDV(V1)
```

创建名为 TEMP 的新列，其中包含单个值，即 V1 列内容的标准差。

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

创建名为 TEMP 的新列，其中包含单个值，即 V1、V2 和 V3 列内容的标准差。

TEMP = STDV(V1[1:5]:V4)
创建名为 TEMP 的新列，其中包含单个值，即 V1 至 V4 列第 1-5 行单元格的标准差。
TEMP = STDV(V1:V3, COL)
创建名称分别为 TEMP、VX 和 VY 的三个新列。TEMP 列中的单个值是 V1 列内容的标准差，VX 列中的单个值是 V2 列内容的标准差，而 VY 列中的单个值是 V3 列内容的标准差。
TEMP = STDV(V1[10:50]:V3,COL)
创建名称分别为 TEMP、VX 和 VY 的三个新列，其中每个列都包含单个值。TEMP 列中的值是 V1 列第 10-50 行单元格的标准差，VX 列中的值是 V2 列第 10-50 行单元格的标准差，而 VY 列中的值是 V3 列第 10-50 行单元格的标准差。
TEMP = STDV(V1:V3, ROW)
创建名为 TEMP 的新列，其中每个单元格条目都是 V1、V2 和 V3 列中相应行的标准差。
TEMP = STDV(V1[1:5]:V3,ROW)
创建名为 TEMP 的新列，其中 1-5 行的单元格包含 V1 到 V3 列中相应行的标准差。TEMP 中的其他单元格为空。

## 相关函数

函数	描述
VAR	计算一系列单元格的方差

## STRING\_CONCAT 宏

STRING\_CONCAT 宏在 IBM Campaign 和 IBM Interact 中都可用。

### 语法

STRING\_CONCAT(string1, string2, ... stringN)

### 参数

string

要并置的 ASCII 文本字符串。此参数可以用引号括起来的 ASCII 文本、文本列、包含文本的单元格范围或求值为上述任何类型的表达式。要了解 string 的格式定义（与 data 相同），请参阅适用于您的产品的『使用宏』一章中的『宏函数参数』一节。

### 描述

STRING\_CONCAT 将所指定数据范围内的 ASCII 文本值并置。它针对每个输入列返回一个新列，其中每个新列都包含 strings 的相应行中的字符串并置后的结果。这种逐行计算会针对每一行生成一个结果，直至处理完最短列中的最后一个值为止。

注：每个结果字符串的总宽度不能超过 255 个字符。

IBM Interact 还支持以下语法：

STRING\_CONCAT( string1 , string2 , ... stringN )

例如，STRING\_CONCAT('a', 'b', 'c', 'd') 有效。

## 示例

TEMP = STRING_CONCAT("house", "boat")
创建名为 TEMP 的新列，其中包含 ASCII 文本字符串"houseboat"。
TEMP = STRING_CONCAT(V1, ".")
创建名为 TEMP 的新列，其中每一行都包含 V1 列相应行中的 ASCII 文本字符串追加一个句点。
TEMP = STRING_CONCAT(V1, V2)
创建名为 TEMP 的新列，其中每一行都包含 V1 列中的 ASCII 文本字符串与 V2 列中的文本字符串并置后的结果。
TEMP = STRING_CONCAT(V1:V3, V4:V6)
创建名称分别为 TEMP、VX 和 VY 的三个新列。TEMP 列中的值是 V1 和 V4 列的相应行中的字符串并置后的结果，VX 列的值是 V2 和 V5 列的相应行中的字符串并置后的结果，而 VY 列的值是 V3 和 V6 列的相应行中的字符串并置后的结果。
TEMP = STRING_CONCAT(V1[5:10]:V2, V3:V4)
创建名称分别为 TEMP 和 VX 的两个新列。TEMP 列中的值是 V1 列第 5-10 行中的字符串与 V3 列第 1-6 行并置后的结果。VX 中的值是 V2 列第 5-10 行中的字符串与 V4 列第 1-6 行并置后的结果。
TEMP = STRING_CONCAT('a', 'b', 'c', 'd')
创建名为 TEMP 的新列，其中包含 ASCII 文本字符串"abcd"。

## 相关函数

函数	描述
STRING_HEAD	返回所指定数据范围内每个字符串的前 n 个字符
STRING_LENGTH	返回所指定数据范围内每个字符串的长度
STRING_SEG	返回两个指定下标之间的字符串段
STRING_TAIL	返回所指定数据范围内每个字符串的最后 n 个字符

---

## STRING\_HEAD 宏

STRING\_HEAD 宏仅在 IBM Campaign 中可用。

### 语法

STRING\_HEAD(num\_chars, data)

### 参数

num\_chars

要从 data 中的每个字符串开头开始返回的字符数。此值必须是大于零的正整数。

data

ASCII 文本字符串值。此参数可以用引号括起来的 ASCII 文本、文本列、包含文本的单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

## 描述

STRING\_HEAD 返回所指定数据范围内每个字符串值中的前 num\_chars 个字符。如果 num\_chars 大于文本字符串中的字符数，那么会使用空字符"\0"填充剩余字符。

## 示例

TEMP = STRING_HEAD(3, "JAN 15, 1997")
创建名为 TEMP 的新列，其中包含 ASCII 文本字符串"JAN"。
TEMP = STRING_HEAD(10, "Pressure")
创建名为 TEMP 的新列，其中包含 ASCII 文本字符串"Pressure"。
TEMP = STRING_HEAD(5, V1)
创建名为 TEMP 的新列，其中包含列 V1 中每个字符串的前 5 个字符。
TEMP = STRING_HEAD(1, V1:V3)
创建名称分别为 TEMP、VX 和 VY 的三个新列。TEMP 列中的值是 V1 列相应行中的字符串的第 1 个字符，VX 列的值是 V2 列相应行中的字符串的第 1 个字符，而 VY 列的值是 V3 列相应行中的字符串的第 1 个字符。
TEMP = STRING_HEAD(12, V4[1:50]:V6]
创建名称分别为 TEMP、VX 和 VY 的三个新列。TEMP 列中的值是 V1 列第 1-50 行中的字符串的前 12 个字符，VX 列的值是 V2 列第 1-50 行中的字符串的前 12 个字符，而 VY 列的值是 V3 列第 1-50 行中的字符串的前 12 个字符。

## 相关函数

函数	描述
STRING_CONCAT	将所指定数据范围内的两个文本字符串并置
STRING_LENGTH	返回所指定数据范围内每个字符串的长度
STRING_SEG	返回两个指定下标之间的字符串段
STRING_TAIL	返回所指定数据范围内每个字符串的最后 n 个字符

---

## STRING\_LENGTH 宏

STRING\_LENGTH 宏仅在 IBM Campaign 中可用。

### 语法

STRING\_LENGTH(data)

### 参数

data

要计算长度的 ASCII 文本字符串值。此参数可以用引号括起来的 ASCII 文本、文本列、包含文本的单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

## 描述

STRING\_LENGTH 返回所指定数据范围内每个字符串值的长度。它针对每个输入列返回一个新列，其中每个新列都包含相应文本字符串的长度。

注：如果对包含数字数据的列应用 STRING\_LENGTH，那么将返回零。

## 示例

TEMP = STRING_LENGTH("four") 创建名为 TEMP 的新列，其中包含值 4。
TEMP = STRING_LENGTH(4) 创建名为 TEMP 的新列，其中包含值 0。
TEMP = STRING_LENGTH(V1) 创建名为 TEMP 的新列，其中每个值都是 V1 列相应行中的字符串的长度。
TEMP = STRING_LENGTH(V1:V3) 创建名称分别为 TEMP、VX 和 VY 的三个新列。TEMP 列中的值是 V1 列相应行中的字符串的长度，VX 列的值是 V2 列相应行中的字符串的长度，而 VY 列的值是 V3 列相应行中的字符串的长度。
TEMP = STRING_LENGTH(V4[1:50]:V6) 创建名称分别为 TEMP、VX 和 VY 的三个新列。TEMP 列中的值是 V1 列第 1-50 行中的字符串的长度，VX 列的值是 V2 列第 1-50 行中的字符串的长度，而 VY 列的值是 V3 列第 1-50 行中的字符串的长度。

## 相关函数

函数	描述
STRING_CONCAT	将所指定数据范围内的两个文本字符串并置
STRING_HEAD	返回所指定数据范围内每个字符串的前 n 个字符
STRING_SEG	返回两个指定下标之间的字符串段
STRING_TAIL	返回所指定数据范围内每个字符串的最后 n 个字符

---

## STRING\_PROPER 宏

STRING\_PROPER 宏仅在 IBM Campaign 中可用。

## 语法

STRING\_PROPER(data)

## 参数

data

要转换的字符串值。

## 描述

STRING\_PROPER 按以下方式转换所指定数据范围内的每个字符串值：将第一个字母或者空格字符或符号后面的任何非下划线字母转换为大写，并将所有其他字符转换为小写。它针对每个输入列返回一个新列，其中每个新列都包含相应输入列中的字符串转换后的结果。

## 示例

```
Temp = STRING_PROPER
```

---

## STRING\_SEG 宏

STRING\_SEG 宏仅在 IBM Campaign 中可用。

## 语法

STRING\_SEG(from, to, data)

## 参数

from

相对于字符串开头的偏移字符数，字符串段抽取从此处开始。此值必须是大于零且小于 to 的正整数，否则 STRING\_SEG 将返回空字符串。

to

相对于字符串开头的偏移字符数，字符串段抽取在此处结束。此值必须是大于或等于 from 的正整数。如果 to 等于 from（并且 to 小于或等于字符串长度），那么将返回一个字符。

data

ASCII 文本字符串值。此参数可以用引号括起来的 ASCII 文本、文本列、包含文本的单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

## 描述

STRING\_SEG 返回所指定数据范围内每个字符串值中介于两个索引之间的字符串段。如果 from 大于字符串长度，那么不返回任何内容。如果 to 大于字符串长度，那么返回从 from 开始的所有字符。

## 示例

<pre>TEMP = STRING_SEG(1, 6, "JAN 15, 1997")</pre>
创建名为 TEMP 的新列，其中包含 ASCII 文本字符串"Jan 15"。
<pre>TEMP = STRING_SEG(5, 20, "Pressure")</pre>
创建名为 TEMP 的新列，其中包含 ASCII 文本字符串"sure"。
<pre>TEMP = STRING_SEG(5, 6, V1)</pre>
创建名为 TEMP 的新列，其中包含列 V1 中每个字符串的第 5 个字符和第 6 个字符。
<pre>TEMP = STRING_SEG(10, 20, V1:V3)</pre>
创建名称分别为 TEMP、VX 和 VY 的三个新列。TEMP 列中的值是 V1 列相应行中的字符串的第 10-20 个字符，VX 列的值是 V2 列相应行中的字符串的第 10-20 个字符，而 VY 列的值是 V3 列相应行中的字符串的第 10-20 个字符。
<pre>TEMP = STRING_SEG(5, 10, V4[1:50]:V6]</pre>
创建名称分别为 TEMP、VX 和 VY 的三个新列。TEMP 列中的值是 V1 列第 1-50 行中的字符串的第 5-10 个字符，VX 列的值是 V2 列第 1-50 行中的字符串的第 5-10 个字符，而 VY 列的值是 V3 列第 1-50 行中的字符串的第 5-10 个字符。

## 相关函数

函数	描述
STRING_CONCAT	将所指定数据范围内的两个文本字符串并置
STRING_HEAD	返回所指定数据范围内每个字符串的前 n 个字符
STRING_LENGTH	返回所指定数据范围内每个字符串的长度
STRING_TAIL	返回所指定数据范围内每个字符串的最后 n 个字符

---

## STRING\_TAIL 宏

STRING\_TAIL 宏仅在 IBM Campaign 中可用。

### 语法

```
STRING_TAIL(num_chars, data)
```

### 参数

num\_chars

要从 data 中的每个字符串末尾开始返回的字符数。此值必须是大于零的正整数。

data

ASCII 文本字符串值。此参数可以用引号括起来的 ASCII 文本、文本列、包含文本的单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

## 描述

STRING\_TAIL 返回所指定数据范围内每个字符串值中的最后 num\_chars 个字符。所有字符串值都会以空字符"\0"填充，直至达到最大字符串长度为止。然后，返回每个字符串中的最后 num\_chars 个字符。如果 num\_chars 大于文本字符串中的字符数，那么返回整个文本字符串。

## 示例

TEMP = STRING_TAIL(3, "JAN 15, 1997")
创建名为 TEMP 的新列，其中包含 ASCII 文本字符串"997"。
TEMP = STRING_TAIL(10, "Pressure")
创建名为 TEMP 的新列，其中包含 ASCII 文本字符串"Pressure"。
TEMP = STRING_TAIL(5, V1)
创建名为 TEMP 的新列，其中包含列 V1 中每个字符串的最后 5 个字符。
TEMP = STRING_TAIL(1, V1:V3)
创建名称分别为 TEMP、VX 和 VY 的三个新列。TEMP 列中的值是 V1 列相应行中的字符串的最后 1 个字符，VX 列的值是 V2 列相应行中的字符串的最后 1 个字符，而 VY 列的值是 V3 列相应行中的字符串的最后 1 个字符。
TEMP = STRING_TAIL(12, V4[1:50]:V6)
创建名称分别为 TEMP、VX 和 VY 的三个新列。TEMP 列中的值是 V1 列第 1-50 行中的字符串的最后 12 个字符，VX 列的值是 V2 列第 1-50 行中的字符串的最后 12 个字符，而 VY 列的值是 V3 列第 1-50 行中的字符串的最后 12 个字符。

## 相关函数

函数	描述
STRING_CONCAT	将所指定数据范围内的两个文本字符串并置
STRING_HEAD	返回所指定数据范围内每个字符串的前 n 个字符
STRING_LENGTH	返回所指定数据范围内每个字符串的长度
STRING_SEG	返回两个指定下标之间的字符串段

---

## SUBSTR 或 SUBSTRING 宏

SUBSTR 或 SUBSTRING 宏在 IBM Campaign 和 IBM Interact 中都可用。

### 语法

SUBSTR(string\_value, start\_pos[, nchars])、SUBSTR(string\_value FROM start\_pos[ FOR nchars])、SUBSTRING(string\_value, start\_pos[, nchars]) 或 SUBSTRING(string\_value FROM start\_pos[ FOR nchars])

## 参数

`string_value`

要从中获取子串的字符串。

`start_pos`

每个所抽取子串的起始字符。

`nchars`

要抽取的字符数（必须大于或等于 0）。如果未提供此值，那么将抽取 `string_value` 中的所有剩余字符。

## 描述

SUBSTR 或 SUBSTRING 从字符串中的 `start_pos` 处开始抽取 `nchars` 个字符。如果省略 `nchars`，那么 SUBSTR 和 SUBSTRING 将抽取从 `start_pos` 开始直到字符串末尾的字符。尾部空格将被自动截断。为了避免发生语法错误，请确保以逗号和空格分隔数字值，如以下示例所示。

**要点：**IBM Interact 仅支持下列格式：SUBSTR(`string_value`, `start_pos`[, `nchars`]) 或 SUBSTRING(`string_value`, `start_pos`[, `nchars`])

## 示例

SUBSTR SUBSTR 返回结果	("abcdef" FROM 1 FOR 2) ("abcdef", 1, 2) 'ab'
SUBSTR SUBSTR 返回结果	("abcdef" FROM -2 FOR 4) ("abcdef", -2, 4) 'a'
SUBSTR SUBSTR 返回结果	("abcdef" FROM 3) ("abcdef", 3) 'cdef'

---

## SUM 宏

SUM 宏在 IBM Campaign 和 IBM Interact 中都可用。

## 语法

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

## 参数

`data`

要计算总和的数字值。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。要了解 `data` 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

`keyword`

这个可选关键字确定如何对输入数据范围执行计算。请选择下列其中一项：

ALL - 对 `data` 中的所有单元格执行计算（缺省值）

COL - 对 data 的每一列单独执行计算

ROW - 对 data 的每一行单独执行计算

有关在 IBM Campaign 中使用关键字的更多详细信息，请参阅第 6 页的『格式规范』。

注：许多宏函数使用关键字参数 {ALL | COL | ROW}。这些关键字在 IBM Campaign 中不适用，因为输入数据始终为单个列或字段。这个宏的行为始终像是已指定 COL 关键字一样。因此，使用 IBM Campaign 时，无需指定这些关键字。

## 描述

SUM 计算所指定数据范围内所有单元格的总和。它返回单个列。

注：SUM 与 TOTAL 宏函数等同。

## 示例

TEMP = SUM(3)创建名为 TEMP 的新列，其中包含值 3。
TEMP = SUM((COLUMN(3, 5, 1)) 创建名为 TEMP 的新列，其中包含值 9。
TEMP = SUM(V1) 创建名为 TEMP 的新列，其中包含单个值，即 V1 列内容的总和。
TEMP = SUM(V1:V3) 创建名为 TEMP 的新列，其中包含单个值，即 V1、V2 和 V3 列内容的总和。
TEMP = SUM(V1[1:5]:V4) 创建名为 TEMP 的新列，其中包含单个值，即 V1 至 V4 列第 10-20 行单元格的总和。
TEMP = SUM(V1:V3, COL) 创建名称分别为 TEMP、VX 和 VY 的三个新列。TEMP 列中的单个值是 V1 列内容的总和，VX 列中的单个值是 V2 列内容的总和，而 VY 列中的单个值是 V3 列内容的总和。
TEMP = SUM(V1[1:5]:V3, COL) 创建名称分别为 TEMP、VX 和 VY 的三个新列，其中每个列都包含单个值。TEMP 列中的值是 V1 列第 1-5 行单元格的总和，VX 列中的值是 V2 列第 1-5 行单元格的总和，而 VY 列中的值是 V3 列第 1-5 行单元格的总和。
TEMP = SUM(V1:V3, ROW) 创建名为 TEMP 的新列，其中每个单元格条目都是 V1、V2 和 V3 列中相应行的总和。
TEMP = SUM(V1[1:5]:V3, ROW) 创建名为 TEMP 的新列，其中 1-5 行的单元格包含 V1 到 V3 列中相应行的总和。TEMP 中的其他单元格为空。

## 相关函数

函数	描述
AVG 或 MEAN	计算一系列单元格的平均值或算术平均值

---

## TAN 宏

TAN 宏仅在 IBM Campaign 中可用。

### 语法

TAN(data [, units\_keyword])

### 参数

data

要计算正切值的数字值。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

units\_keyword

这个可选关键字确定输入值和结果是解释为度数还是弧度数。请选择下列其中一项：

RADIAN - 以弧度为单位执行计算（缺省情况）

DEGREE - 以度为单位执行计算

如果未指定此参数，那么缺省情况下将以弧度为单位。（要从弧度转换为度，请除以 PI 并乘以 180。）

有关在 IBM Campaign 中使用关键字的更多详细信息，请参阅第 6 页的『格式规范』。

### 描述

TAN 计算所指定数据范围内值的正切值。它针对每个输入列返回一个新列，其中每个新列都包含相应输入列中数字的正切值。

### 示例

TEMP = TAN(PI/4)、TEMP = TAN(PI/4, 0) 或 TEMP = TAN(PI/4, RADIAN) 创建名为 TEMP 的新列，其中包含值 1。
TEMP = TAN(V1) 创建名为 TEMP 的新列，其中每个值都是 V1 列内容的正切值（以弧度计）。
TEMP = TAN(V1:V3, 1) 或 TEMP = TAN(V1:V3, DEGREE) 创建名称分别为 TEMP、VX 和 VY 的三个新列。TEMP 列中的值是 V1 列内容的正切值，VX 列的值是 V2 列内容的正切值，而 VY 列的值是 V3 列内容的正切值。所有值都以度为单位。

```
TEMP = TAN(V1[1:5]:V2)
```

创建名称分别为 TEMP 和 VX 的两个新列，每个列在第 1-5 行都有值（其他单元格为空）。列 TEMP 中的值是列 V1 中相应行的正切值，而列 VX 中的值是列 V2 中相应行的正切值。所有值都以弧度为单位。

## 相关函数

函数	描述
ATAN	计算所指定数据范围中内容的反正切值
COS	计算所指定数据范围中内容的余弦值
COT	计算所指定数据范围中内容的余切值
SIN	计算所指定数据范围中内容的正弦值
TANH	计算所指定数据范围中内容的双曲正切值

---

## TANH 宏

TANH 宏仅在 IBM Campaign 中可用。

### 语法

```
TANH(data [, units_keyword])
```

### 参数

data

要计算双曲正切值的数字值。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

units\_keyword

这个可选关键字确定输入值和结果是解释为度数还是弧度数。请选择下列其中一项：

RADIAN - 以弧度为单位执行计算（缺省情况）

DEGREE - 以度为单位执行计算

如果未指定此参数，那么缺省情况下将以弧度为单位。（要从弧度转换为度，请除以 PI 并乘以 180。）

有关在 IBM Campaign 中使用关键字的更多详细信息，请参阅第 6 页的『格式规范』。

### 描述

TANH 计算所指定数据范围内值的双曲正切值。它针对每个输入列返回一个新列，其中每个新列都包含相应输入列中数字的双曲正切值。数字的双曲正切值的计算方式如下：

$$\tanh(x) = \frac{\sinh(x)}{\cosh(x)}$$

注：当值  $x$  过大时，将会返回溢出错误。 $\tanh(x)$  超过最大 32 位浮点值时就会发生这种情况。如果  $\cosh(x)$  为零，那么  $\text{TANH}$  返回最大 32 位浮点值。

### 示例

TEMP = TANH(PI)、TEMP = TANH(PI, 0) 或 TEMP = TANH(PI, RADIAN)
创建名为 TEMP 的新列，其中包含值 1。
TEMP = TANH(V1)
创建名为 TEMP 的新列，其中每个值都是 V1 列内容的双曲正切值（以弧度计）。
TEMP = TANH(V1:V3, 1) 或 TEMP = TANH(V1:V3, DEGREE)
创建名称分别为 TEMP、VX 和 VY 的三个新列。TEMP 列中的值是 V1 列内容的双曲正切值，VX 列的值是 V2 列内容的双曲正切值，而 VY 列的值是 V3 列内容的双曲正切值。所有值都以度为单位。
TEMP = TANH(V1[1:5]:V2)
创建名称分别为 TEMP 和 VX 的两个新列，每个列在第 1-5 行都有值（其他单元格为空）。列 TEMP 中的值是列 V1 中相应行的双曲正切值，而列 VX 中的值是列 V2 中相应行的双曲正切值。所有值都以弧度为单位。

### 相关函数

函数	描述
ATAN	计算所指定数据范围中内容的反正切值
COSH	计算所指定数据范围中内容的双曲余弦值
COT	计算所指定数据范围中内容的余切值
SINH	计算所指定数据范围中内容的双曲正弦值
TAN	计算所指定数据范围中内容的正切值

---

## TOTAL 宏

TOTAL 宏在 IBM Campaign 和 IBM Interact 中都可用。

### 语法

TOTAL(data [, keyword])

### 参数

data

要计算总和的数字值。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

keyword

这个可选关键字确定如何对输入数据范围执行计算。请选择下列其中一项：

ALL - 对 data 中的所有单元格执行计算（缺省值）

COL - 对 data 的每一列单独执行计算

ROW - 对 data 的每一行单独执行计算

有关在 IBM Campaign 中使用关键字的更多详细信息，请参阅第 6 页的『格式规范』。

注：许多宏函数使用关键字参数 {ALL | COL | ROW}。这些关键字在 IBM Campaign 中不适用，因为输入数据始终为单个列或字段。这个宏的行为始终像是已指定 COL 关键字一样。因此，使用 IBM Campaign 时，无需指定这些关键字。

## 描述

TOTAL 计算所指定数据范围内所有单元格的总和。

注：TOTAL 与 SUM 宏函数等同。

## 示例

TEMP = TOTAL(3) 创建名为 TEMP 的新列，其中包含值 3。
TEMP = TOTAL((COLUMN(3, 5, 1))) 创建名为 TEMP 的新列，其中包含值 9。
TEMP = TOTAL(V1) 创建名为 TEMP 的新列，其中包含单个值，即 V1 列内容的总和。
TEMP = TOTAL(V1:V3) 创建名为 TEMP 的新列，其中包含单个值，即 V1、V2 和 V3 列内容的总和。
TEMP = TOTAL(V1[1:5]:V4) 创建名为 TEMP 的新列，其中包含单个值，即 V1 至 V4 列第 10-20 行单元格的总和。
TEMP = TOTAL(V1:V3, COL) 创建名称分别为 TEMP、VX 和 VY 的三个新列。TEMP 列中的单个值是 V1 列内容的总和，VX 列中的单个值是 V2 列内容的总和，而 VY 列中的单个值是 V3 列内容的总和。
TEMP = TOTAL(V1[1:5]:V3, COL) 创建名称分别为 TEMP、VX 和 VY 的三个新列，其中每个列都包含单个值。TEMP 列中的值是 V1 列第 1-5 行单元格的总和，VX 列中的值是 V2 列第 1-5 行单元格的总和，而 VY 列中的值是 V3 列第 1-5 行单元格的总和。

TEMP = TOTAL(V1:V3, ROW)
创建名为 TEMP 的新列，其中每个单元格条目都是 V1、V2 和 V3 列中相应行的总和。
TEMP = TOTAL(V1[1:5]:V3, ROW)
创建名为 TEMP 的新列，其中 1-5 行的单元格包含 V1 到 V3 列中相应行的总和。TEMP 中的其他单元格为空。

## 相关函数

函数	描述
AVG 或 MEAN	计算一系列单元格的平均值或算术平均值

## TRUNCATE 宏

TRUNCATE 宏仅在 IBM Campaign 中可用。

### 语法

TRUNCATE(data)

### 参数

data

要截断的数字值。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

### 描述

TRUNCATE 计算所指定数据范围内每个值的整数部分。它针对每个输入列返回一个新列，其中每个新列都包含相应输入列中数字的整数（非小数）部分。

注：FRACTION 宏函数与 TRUNCATE 宏函数互补，这两者之和等于原始值。

### 示例

TEMP = TRUNCATE(4.3)
创建名为 TEMP 的新列，其中包含值 4。
TEMP = TRUNCATE(2.9)
创建名为 TEMP 的新列，其中包含值 -2。
TEMP = TRUNCATE(V1)
创建名为 TEMP 的新列，其中每个值都是 V1 列内容的小数部分。
TEMP = TRUNCATE(V1:V3)
创建名称分别为 TEMP、VX 和 VY 的三个新列。TEMP 列中的值是 V1 列截断后的部分，VX 列的值是 V2 列截断后的部分，而 VY 列的值是 V3 列截断后的部分。

TEMP = TRUNCATE(V1[10:20])
创建名为 TEMP 的新列，其中，前 11 个单元格包含 V1 列第 10-20 行中的值截断后的部分。TEMP 中的其他单元格为空。
TEMP = TRUNCATE(V1[50:99]:V2)
创建名称分别为 TEMP 和 VX 的两个新列，每个列在第 1-50 行都有值（其他单元格为空）。列 TEMP 中的值是列 V1 中各行截断后的部分，而列 VX 中的值是列 V2 中各个值截断后的部分。

## 相关函数

函数	描述
CEILING	计算所指定数据范围内每个值的向上取整
FLOOR	计算所指定数据范围内每个值的向下取整
FRACTION	返回所指定数据范围内每个值的小数部分

---

## UPPER 宏

UPPER 宏在 IBM Campaign 和 IBM Interact 中都可用。

### 语法

UPPER(data)

### 参数

data

要转换为大写的字符串值。

### 描述

UPPER 将所指定数据范围内的每个字符串值转换为大写。它针对每个输入列返回一个新列，其中每个新列都包含相应输入列中的大写字符串。

### 示例

Temp = UPPER "gold"
创建名为 Temp 的新列，其中包含 "GOLD"。
TEMP = UPPER( "jan 15, 1997")
创建名为 TEMP 的新列，其中包含 ASCII 文本字符串 "JAN 15, 1997"。
TEMP = UPPER( "Pressure")
创建名为 TEMP 的新列，其中包含 ASCII 文本字符串 "PRESSURE"。
TEMP = UPPER(V1)
创建名为 TEMP 的新列，其中包含列 V1 中每个字符串的大写字符。

## VARIANCE 宏

VARIANCE 宏仅在 IBM Campaign 中可用。

### 语法

VARIANCE(data [, keyword])

### 参数

data

要计算方差的数字值。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

keyword

这个可选关键字确定如何对输入数据范围执行计算。请选择下列其中一项：

ALL - 对 data 中的所有单元格执行计算（缺省值）

COL - 对 data 的每一列单独执行计算

ROW - 对 data 的每一行单独执行计算

有关在 IBM Campaign 中使用关键字的更多详细信息，请参阅第 6 页的『格式规范』。

注：许多宏函数使用关键字参数 {ALL | COL | ROW}。这些关键字在 **IBM Campaign** 中不适用，因为输入数据始终为单个列或字段。这个宏的行为始终像是已指定 COL 关键字一样。因此，使用 **IBM Campaign** 时，无需指定这些关键字。

### 描述

VARIANCE 计算所指定数据范围内所有值的方差。方差是标准差的平方。方差的计算方式如下：

$$\frac{1}{n-1} \sum_{j=1}^n (x_j - \text{mean})^2$$

其中， $x$  是样本， $n$  是样本数，而  $\text{mean}$  是分布的平均值。

注：如果样本数  $n = 1$ ，那么 VARIANCE 将返回错误。

### 示例

```
TEMP = VARIANCE(V1)
```

创建名为 TEMP 的新列，其中包含单个值，即 V1 列内容的方差。

TEMP = VARIANCE(V1:V3)
创建名为 TEMP 的新列，其中包含单个值，即 V1、V2 和 V3 列内容的方差。
TEMP = VARIANCE(V1[10:20])
创建名为 TEMP 的新列，其中包含单个值，即 V1 列第 10-20 行单元格的方差。
TEMP = VARIANCE(V1[1:5]:V4)
创建名为 TEMP 的新列，其中包含单个值，即 V1 至 V4 列第 1-5 行单元格的方差。
TEMP = VARIANCE(V1:V3, COL)
创建名称分别为 TEMP、VX 和 VY 的三个新列。TEMP 列中的单个值是 V1 列内容的方差，VX 列中的单个值是 V2 列内容的方差，而 VY 列中的单个值是 V3 列内容的方差。
TEMP = VARIANCE_(V1[1:5]:V3, COL) 或 TEMP = VARIANCE(V1[1:5]:V3[1:5], COL)
创建名称分别为 TEMP、VX 和 VY 的三个新列，其中每个列都包含单个值。TEMP 列中的值是 V1 列第 1-5 行单元格的方差，VX 列中的值是 V2 列第 1-5 行单元格的方差，而 VY 列中的值是 V3 列第 1-5 行单元格的方差。
TEMP = VARIANCE(V1:V3, ROW)
创建名为 TEMP 的新列，其中每个单元格条目是 V1、V2 和 V3 列中的相应行的方差。
TEMP = VARIANCE(V1[1:5]:V3,ROW) 或 TEMP = VARIANCE(V1[1:5]:V3[1:5], ROW)
创建名为 TEMP 的新列，其中 1-5 行的单元格包含 V1 到 V3 列中相应行的方差。TEMP 中的其他单元格为空。

## WEEKDAY 宏

WEEKDAY 宏仅在 IBM Campaign 中可用。

### 语法

WEEKDAY(data [, conversion\_keyword])

### 参数

data

要转换为表示一周中第几天 (1-7) 的数字值的 ASCII 文本日期。此参数可以用引号括起来的 ASCII 文本、文本列、包含文本的单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义，请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

conversion\_keyword

这个可选关键字指定如何解释日期和时间的文本格式。请选择下列其中一项：

- 1 - mm/dd/yy (缺省值)
- 2 - dd-mmm-yy
- 3 - mm/dd/yy hh:mm

如果未指定此参数，那么缺省情况下将使用 1。

## 描述

WEEKDAY 通过使用指定的格式来转换日期和时间，将所指定数据范围内的文本值转换为表示一周中第几天的数字值。数字 0 表示星期日，1 表示星期一，依此类推，直到 6 表示星期六。如果无法使用指定的 conversion\_keyword 解析文本字符串，那么 WEEKDAY 将返回错误。

## 示例

TEMP = WEEKDAY("1/1/95")
创建名为 TEMP 的新列，其中包含数字 0（1995 年 1 月 1 日是星期日）。
TEMP = WEEKDAY(V1, 2)
创建名为 TEMP 的新列，其中包含表示列 V1 中文本字符串是一周中第几天的数字。列 V1 中所有文本字符串的格式都应该为 dd-mmm-yy（否则，将返回 ???）。
TEMP = WEEKDAY(V1:V3, 3)
创建名称分别为 TEMP、VX 和 VY 的三个新列。列 TEMP 包含表示列 V1 中文本字符串是一周中第几天的数字。列 VX 包含表示列 V2 中文本字符串是一周中第几天的数字。列 VY 包含表示列 V3 中文本字符串是一周中第几天的数字。列 V1 - V3 中所有文本字符串的格式都应该为 mm/dd/yy hh:mm（否则，将返回 ???）。
TEMP = WEEKDAY(V1[10:20]:V2, 10)
创建名称分别为 TEMP 和 VX 的两个新列。列 TEMP 包含表示 V1 列第 10-20 行文本字符串是一周中第几天的数字。列 VX 包含表示 V2 列第 10-20 行文本字符串是一周中第几天的数字。所有文本字符串的格式都应该为 mm/dd/yy（否则，将返回 ???）。

## 相关函数

函数	描述
NUMBER	将表示时间和日期的 ASCII 文本字符串转换为数字值

---

## WEEKDAYOF 宏

WEEKDAYOF 宏仅在 IBM Campaign 中可用。

### 语法

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

### 参数

date\_string

表示有效日期的文本。

input\_format

下表中的某个关键字，用于指定 date\_string 的日期格式。

## 描述

WEEKDAYOF 以介于 0-6 (0 表示星期日, 1 表示星期一, 依此类推) 的数字形式返回 date\_string 所指定日期是一周中的第几天。如果未提供 input\_format, 那么将使用缺省关键字 DELIM\_M\_D\_Y。

## 示例

WEEKDAYOF("08312000", MMDDYYYY) 返回数字 4, 因为星期四是一周的第 4 天。

注: 请参阅 第 46 页的『DATE』, 以获取有效日期格式的更多相关信息。

## 相关函数

函数	描述
DAYOF	以数字形式返回一个月中的日子。
MONTHOF	以数字形式返回一年中的月份。
YEAROF	以数字形式返回年份。

---

## XOR 宏

XOR 宏在 IBM Campaign 和 IBM Interact 中可用。

## 语法

```
data1 XOR data2
```

## 参数

data1

要与 data2 中的值进行按位 XOR 运算的非负整数。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。要了解 data 的格式定义, 请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

data2

要与 data1 中的值进行按位 XOR 运算的非负整数。此参数可以是常量值、列、单元格范围或求值为上述任何类型的表达式。除非 data2 是常量, 否则 data2 中的列数必须与 data1 中的列数相等。要了解 data 的格式定义, 请参阅本指南中有关 IBM 产品的章节中的『宏函数参数』一节。

## 描述

XOR 在两个所指定数据范围之间执行按位 XOR 运算。它针对每个输入列返回一个新列, 其中每个新列都包含 data1 的相应列与 data2 的相应列进行按位 XOR 运算的结果 (即, data1 的第一列与 data2 的第一列进行按位 XOR 运算, 第二列与第二列进行按位 XOR 运算, 依此类推)。

如果 data2 是常量，那么 data1 中的每个值都与该值进行按位 XOR 运算。如果 data2 包含一个或多个列，那么将在 data1 中的一列与 data2 中的一列之间逐行执行计算。data1 的第一行与 data2 的第一个行值进行按位 XOR 运算，第二行与第二行进行按位 XOR 运算，依此类推。这种逐行计算会针对每一行生成一个结果，直至处理完最短列中的最后一个值为止。

注：这个宏函数的精度限制为小于  $2^{24}$  的整数值。不允许使用负值。

## 示例

TEMP = 3 XOR 7
创建名为 TEMP 的新列，其中包含值 4 (011 与 111 的按位 XOR 运算结果为 100)。
TEMP = V1 XOR 8
创建名为 TEMP 的新列，其中每个值都是列 V1 的内容与二进制值 1000 进行按位 XOR 运算的结果。
TEMP = V1 XOR V1
创建名为 TEMP 的新列，其中包含的值全为零（每个值与其自身执行 XOR 运算会生成零值）。
TEMP = V1 XOR V2
创建名为 TEMP 的新列，其中每个值都是列 V1 的行值与列 V2 的相应行值进行按位 XOR 运算的结果。
TEMP = V1:V3 XOR V4:V6
创建名称分别为 TEMP、VX 和 VY 的三个新列。列 TEMP 包含 V1 中的值与列 V4 的相应行值进行按位 XOR 运算的结果。列 VX 包含列 V2 与 V5 进行按位 XOR 运算所生成的值。列 VY 包含列 V3 与 V6 进行按位 XOR 运算所生成的值。
TEMP = V1[10:20] XOR V2 或 TEMP = V1[10:20] XOR V2[1:11]
创建名为 TEMP 的新列，其中，前 11 个单元格包含 V1 列第 10-20 行的值与 V2 列第 1-11 行的值进行按位 XOR 运算的结果。TEMP 中的其他单元格为空。

## 相关函数

函数	描述
BIT_AND	计算两个指定数据范围之间的按位 AND 运算结果
BIT_NOT	计算所指定数据范围中内容的按位 NOT 运算结果
BIT_OR	计算两个指定数据范围之间的按位 OR 运算结果

## YEAROF 宏

YEAROF 宏仅在 IBM Campaign 中可用。

## 语法

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

## 参数

date\_string

表示有效日期的文本。

input\_format

下表中的某个关键字，用于指定 date\_string 的日期格式。

## 描述

YEAROF 以数字形式返回 date\_string 所指定日期的年份。如果未提供 input\_format，那么将使用缺省关键字 DELIM\_M\_D\_Y。

## 示例

YEAROF("31082000", DDMMYYYY) 将返回数字 2000。

有关有效日期格式的其他信息，请参阅 第 46 页的『DATE』。

## 相关函数

函数	描述
DAYOF	以数字形式返回一个月中的日子。
MONTHOF	以数字形式返回一年中的月份。
WEEKDAYOF	以数字形式返回某个日期是一周中的第几天。



---

## 在与 IBM 技术支持联系之前

如果您遇到无法通过查阅文档解决的问题，那么贵公司的指定支持联系人可致电 IBM 技术支持中心。使用这些准则来确保您的问题得以有效且成功地解决。

如果您不是贵公司的指定支持联系，请与 IBM 管理员联系以了解相关信息。

注：技术支持不会编写或创建 API 脚本。有关实现 API 产品的帮助，请与 IBM 专业服务联系。

### 要收集的信息

联系 IBM 技术支持前，请收集以下信息：

- 有关问题性质的简短描述。
- 发生问题时看到的详细错误消息。
- 重现该问题的详细步骤。
- 相关的日志文件、会话文件、配置文件和数据文件。
- 关于产品和系统环境的信息，您可以按"系统信息"中所述来获取。

### 系统信息

致电 IBM 技术支持时，可能会要求您提供有关系统环境的信息。

如果问题不妨碍登录，那么可在"关于"页面上获得大部分此类信息，该页面提供有关所安装的 IBM 应用程序的信息。

可以通过选择帮助 > 关于来访问"关于"页面。如果"关于"页面不可访问，请检查位于应用程序安装目录下面的 version.txt 文件。

### IBM 技术支持的联系信息

有关联系 IBM 技术支持中心的方法，请参见 IBM 产品技术支持中心网站：[http://www.ibm.com/support/entry/portal/open\\_service\\_request](http://www.ibm.com/support/entry/portal/open_service_request)。

注：要输入支持请求，您必须使用 IBM 帐户登录。此帐户必须已链接至 IBM 客户编号。要了解有关将您的帐户与 IBM 客户编号相关联的更多信息，请参阅"支持门户网站"上的支持资源 > 授权的软件支持。



---

## 声明

本信息是为在美国提供的产品和服务而编写的。

IBM 可能在其他国家或地区不提供本文中讨论的产品、服务或功能特性。有关您所在区域当前可获得的产品和服务的信息，请向您当地的 IBM 代表咨询。任何对 IBM 产品、程序或服务的引用并非意在明示或暗示只能使用 IBM 的产品、程序或服务。只要不侵犯 IBM 的知识产权，任何同等功能的产品、程序或服务，都可以代替 IBM 产品、程序或服务。但是，评估和验证任何非 IBM 产品、程序或服务，则由用户自行负责。

IBM 可能已拥有或正在申请与本文档内容有关的各项专利。提供本文档并不意味着授予用户使用这些专利的任何许可。您可以用书面形式将许可查询寄往：

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

有关双字节 (DBCS) 信息的许可证查询，请与您所在国家或地区的 IBM 知识产权部门联系，或用书面方式将查询寄往：

Intellectual Property Licensing  
Legal and Intellectual Property Law  
IBM Japan, Ltd.  
19-21, Nihonbashi-Hakozakicho, Chuo-ku  
Tokyo 103-8510, Japan

本条款不适用英国或任何这样的条款与当地法律不一致的国家或地区：International Business Machines Corporation"按现状"提供本出版物，不附有任何种类的（无论是明示的还是暗含的）保证，包括但不限于暗含的有关非侵权、适销和适用于某种特定用途的保证。某些国家或地区在某些交易中不允许免除明示或暗含的保证。因此本条款可能不适用于您。

本信息中可能包含技术方面不够准确的地方或印刷错误。此处的信息将定期更改；这些更改将编入本资料的新版本中。IBM 可以随时对本出版物中描述的产品和/或程序进行改进和/或更改，而不另行通知。

本信息中对任何非 IBM Web 站点的引用都只是为了方便起见才提供的，不以任何方式充当对那些 Web 站点的保证。那些 Web 站点中的资料不是 IBM 产品资料的一部分，使用那些 Web 站点带来的风险将由您自行承担。

IBM 可以按它认为适当的任何方式使用或分发您所提供的任何信息而无需对您承担任何责任。

本程序的被许可方如果要了解有关程序的信息以达到如下目的：(i) 允许在独立创建的程序和其他程序（包括本程序）之间进行信息交换，以及 (ii) 允许对已经交换的信息进行相互使用，请与下列地址联系：

IBM Corporation  
B1WA LKG1  
550 King Street  
Littleton, MA 01460-1250  
U.S.A.

只要遵守适当的条件和条款，包括某些情形下的一定数量的付费，都可获得这方面的信息。

本文档中描述的许可程序及其所有可用的许可资料均由 IBM 依据 IBM 客户协议、IBM 国际程序许可协议或任何同等协议中的条款提供。

此处包含的任何性能数据都是在受控环境中测得的。因此，在其他操作环境中获得的数据可能会有明显的不同。有些测量可能是在开发级的系统上进行的，因此不保证与一般可用系统上进行的测量结果相同。此外，有些测量是通过推算而估算的，实际结果可能会有差异。本文档的用户应当验证其特定环境的适用数据。

涉及非 IBM 产品的信息可从这些产品的供应商、其出版说明或其他可公开获得的资料中获取。IBM 没有对这些产品进行测试，也无法确认其性能的精确性、兼容性或任何其他关于非 IBM 产品的声明。有关非 IBM 产品性能的问题应当向这些产品的供应商提出。

所有关于 IBM 未来方向或意向的声明都可随时变更或收回，而不另行通知，它们仅仅表示了目标和意愿而已。

显示的所有 IBM 价格都是 IBM 建议的最新零售价，可随时更改而不另行通知。经销商的价格可能会有所不同。

本信息包含在日常业务操作中使用的数据和报告的示例。为了尽可能完整地说明这些示例，示例中可能会包括个人、公司、品牌和产品的名称。所有这些名字都是虚构的，若现实生活中实际业务企业使用的名字和地址与此相似，纯属巧合。

版权许可：

本信息包括源语言形式的样本应用程序，这些样本说明不同操作平台上的编程方法。如果是为按照在编写样本程序的操作平台上的应用程序编程接口（API）进行应用程序的开发、使用、经销或分发为目的，您可以任何形式对这些样本程序进行复制、修改、分发，而无需向 IBM 付费。这些示例并未在所有条件下作全面测试。因此，IBM 不能担保或暗示这些程序的可靠性、可维护性或功能。样本程序都是“按现状”提供的，不附有任何种类的保证。对于因使用样本程序而引起的任何损害，IBM 不承担责任。

如果您正以软拷贝格式查看本信息，那么图片和彩色图例可能无法显示。

---

## 商标

IBM、IBM 徽标和 [ibm.com](http://ibm.com) 是 International Business Machines Corp. 在全球许多管辖区域中注册的商标或注册商标。其他产品和服务名称可能是 IBM 或其他公司的商标。Web 上 [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml) 处的“版权和商标”提供了 IBM 商标的当前列表。

---

## 隐私策略和使用条款注意事项

IBM 软件产品（包括作为服务解决方案的软件，即“提供的软件”）可能会使用 cookie 或其他技术来收集产品使用情况信息，以便改善最终用户体验、定制与最终用户的交互或用于其他用途。cookie 是 Web 站点可发送至浏览器的数据，随后可将其存储在您的计算机上作为标识您的计算机的标记。在许多情况下，这些 cookie 不会收集个人信息。如果您要使用的软件产品允许您通过 cookie 或类似技术收集个人信息，我们将在下面告知您具体情况。

根据已部署的配置，此软件产品可能使用会话和持久性 cookie，它们收集各个用户的用户名和其他个人信息以用于会话管理、增强用户可用性或其他使用跟踪或功能性目的。可以禁用这些 cookie，但禁用 cookie 同时也会除去它们所启用的功能。

通过 cookie 和类似技术收集的个人信息由不同的管辖区域监管。如果为此软件产品部署的配置为您（作为客户）提供了通过 cookie 和其他技术从最终用户处收集个人信息的能力，那么您应自行寻求适用于此类数据收集的任何法律的相关法律意见，包括在适当时提供通知和同意文件的任何要求。

IBM 要求客户端 (1) 提供明确、显著的指向客户的 Web 站点使用条款（其中包括指向 IBM 和客户端的数据收集和使用实践的链接）的链接（如隐私政策），(2) 告知访问者 IBM 代表客户将 cookie 和透明 GIF/网络信标存放在访问者的计算机上并说明此类技术的目的，并且 (3) 在法律允许的范围内，在客户或 IBM 代表客户将 cookie 和透明 GIF/网络信标存放在 Web 站点访问者的设备上之前征得 Web 站点访问者的同意。

有关如何使用包括 cookie 在内的各种技术实现这些目的的更多信息，请参阅 IBM“网上隐私声明”(<http://www.ibm.com/privacy/details/us/en>) 中的“Cookie、Web Beacon 和其他技术”部分。



Printed in China