

命令模式

SET variable1 = expression1 [, variable2 = expression2]...

给变量赋值或公式。

参数说明

variable

通过下列方式定义的变量名：

- 1) 通过命令 VARIABLES 定义的变量。
- 2) 可以赋值的提前定义的系统变量。
- 3) 用%包括的环境变量（当前程序或子程序的有效变量）。

expression

数值，或运算公式。

value

数值或字符串（如果多于一个单词应该用引号引起）。

operator

下面的任何一个关键词（每个关键词的前两个字母是有效的）。

PLUS:相加

MINUS:相减或负的（和命令 SUBTRACT 相同）

TIMES:相乘

DIVIDE:相除

POWER:乘方

LOG:10 的对数

SIN:正弦

COS: 余弦

TAN:正切

ASIN:反正弦

ACOS:反余弦

ATAN:反正切

ABS:绝对值

SQRT:开方根

TRUNC:

CHARS:下面字符串中的字符个数

LEFT n:下面字符串中最左端的 n 个字符

RIGHT n:下面字符串中最右端的 n 个字符

SLICE n:从第 n 个字符将线面的字符串分开

ITEM n:将用空格, TAB 键或逗号分开的第 n 项提取出来

CAP:将下列的字符串大写

FILENAME:从指定文件中提取出文件名和扩展名

CLEAN: 将字符串前后的空格删除, 也不打印输出

STR: 将字符串后的空行删除, 将引用转变成单引号内的内容

QUOTE: 将余下的空行也作字符串处理

操作

每个描述的值以字符串的型式保存在相应的变量名下, 即使变量的类型是实数。并不校核数值的范围。

二元计算(如相加)直接从左向右计算, 不允许含有括号。一元计算(如开方根)可以从里向外的嵌套。 在任何的表达式计算前, 所有的变量都可以替代。

引号在调用过程中, 会被忽略, 当用引号引起带有大括号的变量时(例如: "{X}") , 引号同样会被忽略。在变量中储存引号的方法是, 用两个引号将引号引起(例如: SET X="text", 则 X 为 "text") , 或者通过关键字 QUOTE 来实现(例如: SET X=QUOTE "text") 。

无效的计算(如被 0 除或非数值计算)使变量无效。在 IF 语句中 UNDEFINED 与空格相匹配, 因此无效的结果可以用语句 IF {X}="" THEN 来测定。

输出

无

样例

利用变量避免在两个地方赋予相同的值:

```
SET WGT = 1200
```

```
\*** {WGT}-TON CRANE LOAD ***\
```

```
ADD "CRANE LOAD" {WGT} * * *
```

定义 ERROR 避免发生错误时结束程序计算

```
SET ERROR = -1 `避免错误的舱室名称而停止运算。
```

```
TYPE {TANKNAME} FLOODED
```

```
IF {ERROR} <> -1 THEN WAIT `等待操作者读取错误信息。
```

```
SET ERROR = 0 `恢复 ERROR 为正常值
```

表达式:

```
SET N = {N} PLUS 1
```

SET M = MINUS {M}

SET INITIAL = LEFT 1 "{NAME}"

SET A = 3 PLUS 1 DIV 2 `结果为 2，不是 3.5。

SET CHARS "ABC" = 3

SET X = 2 PLUS SQRT ABS -16 TIMES CHARS "ABC" `结果为 18。