

HOPE COMPUTER COMPANY LTD.

Fox pro 2.0
参 考 手 册

刘 湘 王 强 编著
木 子 校



○ 北京希望电脑公司

FoxPro 参 考 手 册

(V2.0)

刘 湘 王 强 编著

木 子 校

北京希望电脑公司

1991.11.

前 言

本书作者长期以来一直从事数据库管理系统的开发和研究工作，本书是作者在充分熟悉了 FoxPro 的全貌的前提下，根据作者在有关 FoxPro 的讲座上的讲义和有关资料整理而成，旨在为 FoxPro 用户和 FoxPro 程序设计人员提供有关 FoxPro 命令和函数的较全面、较仔细の説明，在每个命令或函数的说明之后，均附有该命令或函数的使用实例，是一本具有较好的实用价值的参考书。

全书共分五章：

第一章简明扼要地说明了本书印刷上的约定和本书涉及到的缩写词。

第二章详细介绍了 FoxPro 命令的使用和实例。

第三章详细介绍了 FoxPro 函数的使用和实例。

第四章介绍了 FoxPro 系统内存变量。

第五章说明了自 FoxPro 手册正式付印以来，FoxPro 在界面上和编程语言上的最后修改。

本书的出版得到了北京希望电脑公司的大力协助，在此深深地表示感谢。由于水平有限，不妥之处在所难免，望读者批评指正。

编者

一九九一年十一月

KJS42/06

目 录

前言

第一章 概 述

- 1.1 FoxPro 简介 1
- 1.2 本书印刷上的约定 1
- 1.3 本书使用的缩写词 2

第二章 FoxPro 命令

- 2.1 ? | ?? 4
- 2.2 ??? 5
- 2.3 @...SAY/GET 5
- 2.4 @...BOX 12
- 2.5 @...CLEAR 13
- 2.6 @...FILL 13
- 2.7 @...MEENU 14
- 2.8 @...PROMPT 15
- 2.9 @...TO 16
- 2.10 = 17
- 2.11 ACCEPT 17
- 2.12 ACTIVATE MENU 17
- 2.13. ACTIVATE POPUP 20
- 2.14 ACTIVATE SCREEN 22
- 2.15 ACTIVATE WINDOW 22
- 2.16 APPEND 23
- 2.17 APPEND FROM 24
- 2.18 APPEND FROM ARRAY 25
- 2.19 APPEND MEMO 26
- 2.20 AVERAGE 27
- 2.21 BROWSE 28
- 2.22 CALCULATE 34
- 2.23 CALL 35
- 2.24 CANCEL 36
- 2.25 CHANGE 37

- 2.26 CLEAR 43
- 2.27 CLOSE 45
- 2.28 CLOSE MEMO 46
- 2.29 COMPILE 46
- 2.30 CONTINUE 47
- 2.31 COPY FILE 48
- 2.32 COPY MEMO 48
- 2.33 COPY STRUCTURE 48
- 2.34 COPY STRUCTURE
EXTENDED 49
- 2.35 COPY TO 50
- 2.36 COPY TO ARRAY 51
- 2.37 COUNT 52
- 2.38 CREATE 53
- 2.39 CREATE FROM 53
- 2.40 CREATE LABEL 54
- 2.41 CREATE REPORT 54
- 2.42 CREATE VIEW 54
- 2.43 DEACTIVATE MENU 55
- 2.44 DEACTIVATE POPUP 56
- 2.45 DEACTIVATE WINDOW 58
- 2.46 DECLARE 59
- 2.47 DEFINE BAR 60
- 2.48 DEFINE BOX 62
- 2.49 DEFINE MENU 62
- 2.50 DEFINE PAD 64
- 2.51 DEFINE POPUP 65
- 2.52 DEFINE WINDOW 68
- 2.53 DELETE 71
- 2.54 DELFTE FILE 71
- 2.55 DIMENSION 72
- 2.56 DIRECTORY 73
- 2.57 DISPLAY 74
- 2.58 DISPLAY FILES 74
- 2.59 DISPLAY MEMORY 75

2.60	DISPLAY STATUS.....	76	2.98	MODIFY STRUCTURE	117
2.61	DISPLAY STRUCTURE	77	2.99	MOVE WINDOW	117
2.62	DO	77	2.100	NOTE * &&	118
2.63	DO CASE	78	2.101	ON ERROR	119
2.64	DO WHILE	80	2.102	ON ESCAPE.....	119
2.65	EDIT	81	2.103	ON KEY	121
2.66	EJECT	87	2.104	ON KEY =	122
2.67	EJECT PAGE	87	2.105	ON KEY LABEL.....	123
2.68	ERASE	88	2.106	ON PAD	125
2.69	EXIT	89	2.107	ON PAGE	127
2.70	FILER.....	90	2.108	ON READERROR	127
2.71	FIND	90	2.109	ON SELECTION PAD	128
2.72	FLUSH	91	2.110	ON SELECTION POPUP...	129
2.73	FOR...ENDFOR	91	2.111	PACK	131
2.74	FUNCTION	93	2.112	PARAMETERS	132
2.75	GATHER	94	2.113	PLAY MACRO	132
2.76	GETEXPR	95	2.114	PRINTJOB...	
2.77	GO	96		ENDPRINTJOB	133
2.78	HELP	96	2.115	PRIVATE	134
2.79	HIDE MENU	97	2.116	PROCEDURE	135
2.80	HIDE POPUP	98	2.117	PUBLIC	136
2.81	HIDE WINDOW	100	2.118	QUIT	137
2.82	IF...ENDIF	101	2.119	READ	138
2.83	INDEX	102	2.120	READ MENU	139
2.84	INPUT	103	2.121	RECALL	139
2.85	INSERT	104	2.122	REINDEX	140
2.86	JOIN	104	2.123	RELEASE	140
2.87	KEYBOARD	105	2.124	RELEASE MODULE.....	142
2.88	LABEL	107	2.125	RENAME	142
2.89	LIST	107	2.126	REPLACE	142
2.90	LOAD	108	2.127	REPORT	144
2.91	LOCATE	109	2.128	RESTORE FROM	145
2.92	MENU	110	2.129	RESTORE MACROS	146
2.93	MENU TO	113	2.130	RESTORE SCREEN	146
2.94	MODIFY / COMMAND		2.131	RESTORE WINDOW.....	147
	FILE.....	114	2.132	RESUME	147
2.95	MODIFY LABEL	115	2.133	RETRY	148
2.96	MODIFY MEMO	115	2.134	RETURN	149
2.97	MODIFY REPORT	117	2.135	RUN !	150

2.136	SAVE MACROS	151	2.174	SET DEVICE	171
2.137	SAVE SCREEN	151	2.175	SET DISPLAY.....	171
2.138	SAVE TO	152	2.176	SET DOHISTORY	171
2.139	SAVE WINDOW	152	2.177	SET ECHO	172
2.140	SCAN	153	2.178	SET ESCAPE	172
2.141	SCATTER	154	2.179	SET EXACT.....	173
2.142	SCROOL	155	2.180	SET FIELDS.....	173
2.143	SEEK	156	2.181	SET FILTER	174
2.144	SELECT.....	156	2.182	SET FIXED	174
2.145	SET.....	157	2.183	SET FORMAT.....	174
2.146	SET ALTERNATE	157	2.184	SET FULLPATH.....	175
2.147	SET AUTOSAVE.....	158	2.185	SET FUNCTION	175
2.148	SET BELL.....	158	2.186	SET HEADING	175
2.149	SET BLINK	159	2.187	SET HELP	176
2.150	SET BLOCKSIZE	159	2.188	SET HOURS.....	176
2.151	SET BORDER	159	2.189	SET INDEX	176
2.152	SET BRSTATUS	160	2.190	SET INTENSITY.....	177
2.153	SET CARRY.....	161	2.191	SET LOGERRORS	177
2.154	SET CENTURY	161	2.192	SET MACKKEY.....	177
2.155	SET CLEAR	161	2.193	SET MARGIN	178
2.156	SET CLOCK.....	162	2.194	SET MARK	178
2.157	SET COLOR commands.....	162	2.195	SET MEMOWIDTH	178
2.158	SET COLOR OF.....	163	2.196	SET MESSAGE	179
2.159	SET COLOR OF SCHEME	164	2.197	SET MOUSE	179
2.160	SET COLOR SET	164	2.198	SET NEAR	180
2.161	SET COLOR TO.....	165	2.199	SET NOTIFY	180
2.162	SET COMPATIBLE	165	2.200	SET ODOMTER	180
2.163	SET CONFIRM	166	2.201	SET ORDER.....	181
2.164	SET CONSOLE	166	2.202	SET PATH	181
2.165	SET CURRENCY	167	2.203	SET POINT	182
2.166	SET CURSOR	167	2.204	SET PRINTER.....	182
2.167	SET DATE	167	2.205	SET PROCEDURE.....	182
2.168	SET DEBUG.....	168	2.206	SET RELATION OFF	183
2.169	SET DECIMALS.....	168	2.207	SET RELATION	183
2.170	SET DEFAULT	168	2.208	SET RESOURCE.....	184
2.171	SET DELETED	169	2.209	SET SAFETY	185
2.172	SET DELIMITERS.....	170	2.210	SET SCOREBOARD	185
2.173	SET DEVELOPMENT	170	2.211	SET SEPARATOR	185
			2.212	SET SHADOW.....	186

2.213	SET SPACE	186
2.214	SET STATUS	186
2.215	SET STEP	187
2.216	SET STICKY	187
2.217	SET SYSMENU	187
2.218	SET TALK	188
2.219	SET TOPIC	188
2.220	SET TYPEAHEAD	188
2.221	SET UNIQUE	188
2.222	SET VIEW	189
2.223	SET WINDOW	189
2.224	SHOW MENU	190
2.225	SHOW POPUP	190
2.226	SHOW WINDOW	191
2.227	SKIP	192
2.228	SORT	193
2.229	STORE	194
2.230	SUM	195
2.231	SUSPEND	195
2.232	TEXT...ENDTXET	195
2.233	TOTAL	196
2.234	TYPE	197
2.235	UPDATE	197
2.236	USE	197
2.237	WAIT	198
2.238	ZAP	200

第三章 FoxPro 函数

3.1	&	201
3.2	\$	201
3.3	ABS()	202
3.4	ACOS()	202
3.5	ALIAS()	202
3.6	ALLTRIM()	203
3.7	ASC()	203
3.8	ASIN()	204
3.9	AT()	204
3.10	ATAN()	205
3.11	ATC()	205
3.12	ATCLINE()	206
3.13	ATLINE()	206
3.14	ATN2()	207
3.15	BAR()	207
3.16	BETWEEN()	208
3.17	BOF()	209
3.18	CAPSLOCK()	210
3.19	CDOW()	201
3.20	CEILING()	211
3.21	CHR()	211
3.22	CHRSAW()	212
3.23	CHRTRAN()	212
3.24	CMONTH()	213
3.25	COL()	213
3.26	COS()	213
3.27	CTOD()	214
3.28	CURDIR()	214
3.29	DATE()	215
3.30	DAY()	215
3.31	DBF()	216
3.32	DELETED()	216
3.33	DIFFERENCE()	216
3.34	DISKSPACE()	217
3.35	DMY()	218
3.36	DOW()	218
3.37	DTOC()	218
3.38	DTOR()	219
3.39	DTOS()	219
3.40	EMPTY()	220
3.41	EOF()	220
3.42	ERROR()	221
3.43	EXP()	222
3.44	FCOLSE()	222
3.45	FCOUNT()	223
3.46	FCREATE()	223
3.47	FEOF()	224
3.48	FERROR()	225
3.49	FFLUSH()	226

3.50	FGETS()	226	3.89	LTRIM()	250
3.51	FIELD()	227	3.90	LUPDATE()	250
3.52	FILE()	227	3.91	MAX()	250
3.53	FILTER()	228	3.92	MCOL()	251
3.54	FKLABEL()	228	3.93	MDOWN()	253
3.55	FKMAX()	228	3.94	MDY()	253
3.56	FLOOR()	229	3.95	MEMLINES()	254
3.57	FOPEN()	229	3.96	MEMORY()	255
3.58	FOUND()	230	3.97	MENU()	255
3.59	FPUTS()	231	3.98	MESSAGE()	257
3.60	FREAD()	231	3.99	MIN()	257
3.61	FSEEK()	232	3.100	MLINE()	258
3.62	FSIZE()	233	3.101	MOD()	258
3.63	FULLPATH()	233	3.102	MONTH()	258
3.64	FV()	233	3.103	MROW()	259
3.65	FWRITEE()	234	3.104	NDX()	261
3.66	GETENV()	234	3.105	NUMLOCK()	262
3.67	GETFILE()	234	3.106	OCCURS()	262
3.68	GOMONTH()	235	3.107	ORDER()	262
3.69	HEADER()	235	3.108	OS()	263
3.70	HF()	236	3.109	PAD()	263
3.71	INKEY()	236	3.110	PADC(),PADL(), PADR()	264
3.72	INLIST()	241	3.111	PARAMETERS()	265
3.73	INSMODE()	241	3.112	PAYMENT()	266
3.74	INT()	242	3.113	PCOL()	266
3.75	INALPHA()	242	3.114	PI()	267
3.76	ISCOLOR()	243	3.115	POPUP()	267
3.77	ISDIGIT()	243	3.116	PRINTSTATUS()	268
3.78	ISLOWER()	243	3.117	PROGRAM()	269
3.79	ISUPPER()	243	3.118	PROMPT()	269
3.80	KEY()	244	3.119	PROPER()	271
3.81	LASTKEY()	244	3.120	PROW()	271
3.82	LEFT()	246	3.121	PUTFILE()	271
3.83	LEN()	246	3.122	PV()	272
3.84	LIKE()	247	3.123	RAND()	273
3.85	LINENO()	248	3.124	RAT()	273
3.86	LOG()	249	3.125	RATLINE()	274
3.87	LOG10()	249	3.126	READKEY()	274
3.88	LOWER()	249			

3.127	RECCOUNT()	276
3.128	RECNO()	277
3.129	RECSIZE()	277
3.130	RELATION()	278
3.131	REPLICATE()	279
3.132	RIGHT()	279
3.133	ROUND()	280
3.134	ROW()	280
3.135	RTOD()	280
3.136	RTRIM()	280
3.137	SCHEME()	281
3.138	SCOLS()	281
3.139	SECONDS()	281
3.140	SEEK()	282
3.141	SELECT()	282
3.142	SET()	283
3.143	SIGN()	283
3.144	SIN()	283
3.145	SOUNDEX()	284
3.146	SPACE()	284
3.147	SQRT()	285
3.148	SROWS()	285
3.149	STR()	285
3.150	STRTRAN()	286
3.151	STUFF()	286
3.152	SUBSTR()	287
3.153	SYS()	287
3.154	TAN()	296
3.155	TARGET()	296
3.156	TIME()	297
3.157	TRANSFORM()	297
3.158	TRIM()	297
3.159	TYPE()	298
3.160	UPDATE()	298
3.161	UPPER()	299
3.162	USED()	299
3.163	USER-DEFINE FUNCTIONS (UDFs)	299
3.164	VAL()	302

3.165	VARREAD()	302
3.166	VERSION()	303
3.167	WCOLS()	303
3.168	WEXIST()	304
3.169	WLCOL()	304
3.170	WLROW()	305
3.171	WONTOP()	305
3.172	WOUTPUT()	306
3.173	WROWS()	306
3.174	WVISIBLE()	307
3.175	YEAR()	308

第四章 系統內存變量

4.1	ALIGNMENT	309
4.2	BOX	309
4.3	DBCLICK	309
4.4	INDENT	312
4.5	LMARGIN	312
4.6	PADVANCE	312
4.7	'PAGENO	313
4.8	PBPAGE	314
4.9	PCOLNO	314
4.10	PCOPIES	314
4.11	PDRIVER	315
4.12	PECODE	315
4.13	PEJECT	315
4.14	PEPAGE	316
4.15	PFORM	316
4.16	PLENGTH	317
4.17	PLINENO	317
4.18	PLOFFSET	317
4.19	PPITCH	318
4.20	PQUALITY	318
4.21	PSCODE	318
4.22	PSPACING	319
4.23	PWAIT	319
4.24	RMARGIN	319
4.25	TABS	320

4.26	<i>WRAP</i>	320	5.1.2	编程语言的变化	321
第五章 FoxPro 的最后调整			5.2	操作优化	326
5.1	最后的修改	321	5.3	<i>FoxPro</i> 的 <i>HELP</i>	328
5.1.1	界面的补充修改	321			

第一章 概述

1.1 FoxPro 简介

FoxPro 是目前微机数据库管理系统中功能最强、速度最快、灵活性最大的软件之一。和众多的微机数据库管理系统相比，FoxPro 的优点在于：

- 非编程界面；
- 键盘方式和鼠标方式控制；
- 菜单、窗口、对话框和警告牌；
- 内置文本编辑器；
- 面向对象的报表写作和标签文件生成；
- 比 FoxBASE+ 2.10 增强的200多条语句和比 dBASE IV增强的140多条语句；
- 在 FoxPro 环境中不加修改在运行 FoxBASE+ 和 dBASE III PLUS 程序；
- 支持 dBASE IV的命令和功能；
- 支持数学计算、时间约定和金融计算；
- 支持文件操作功能；
- 等等。

1.2 本书印刷上的约定

为了更好地解释 FoxPro 的命令或函数，本书在印刷上遵循一种固定的格式和一些约定，这些约定描述如下：

大写字母 FoxPro 命令或函数关键字，是 FoxPro 中命令或函数的主体，使用时应准确输入，当命令或函数关键字的长度超过 4 个字符时，FoxPro 支持由命令或函数的前 4 个字符组或的缩写形式。

例如：

COPY, SET, @, REPL(REPLACE)

__XXXX 名字以下划线(_)开始的变量是系统内存变量，在这里，下划线作为系统识别系统内存变量的标志，“XXXX”是系统内存变量名。

例如：

__PLOFFSET, __DBLCLICK

<...> 表示一个应由用户提供实际数据的 FoxPro 命令或函数的一部分，< >符号和任何出现在< >中的小写字母都是用于说明该命令或函数的，它不应该直接输入到命令中。

例如：

TO <memvar>

意味着TO后面应跟着一个特定的内存变量名，直接将 <memvar>

作为命令输入或写到程序中都是错误的。

- (...)
表示 FoxPro 命令或函数中的一个可选部分，即这部分可以要也可以不要，由用户自己选择，当然要这部分和不要这部分，命令或函数的执行结果是不一样的，象 <...> 一样，〔 〕符号和包含在〔 〕中的内容都是用于说明该命令或函数的，用户在使用时不要把出现在其中的小写字母作为命令的一部分直接输入。

例如：

[AT <expN>], [NOWAIT]

用于说明 FoxPro 命令或函数中的选择项，符号“|”不是命令或函数的一部分，但出现在其左侧和右侧的选择项均是命令或函数的一部分，使用时，即可选择其左侧部分，也可选择其右侧部分。

例如：

SET TALK ON | OFF

意味着即可以使用 SET TALK ON，也可以使用 SET TALK OFF。

- (...)
括号作为 FoxPro 函数的一部分，在使用时必须直接出现在函数的关键字之后，括号内是一些有关函数参数的说明，这里所说的括号不包括运算表达式中用来决定运算先后顺序的括号。

例如：

CTOD(<expC>), NUMLOCK({<expL>})

1.3 本书中使用的缩写

在讨论 FoxPro 的使用的各章节中，使用了不少缩写，下面是这些缩写的实际含义。

alias	数据库别名
array	数组名
border string	边界字符串
col	屏幕（窗口）列座标
delimiter	定界符
enhanced	增强色
expC	字符表达式
expD	日期表达式
expL	逻辑表达式
expN	数值表达式
expr	表达式
expr list	表达式列表
field	数据库字段
field list	数据库字段列表

file	文件名
file list	文件名列表
key	关键字
key label	键标
marco name	键盘宏名
memo field	备注型字段
memvar	内存变量
memvar list	内存变量列表
menu list	菜单条列表
menu name	菜单条名
pad name	菜单点名
popup list	弹出式菜单列表
popup name	弹出式菜单名
port	接口
row	屏幕（窗口）行座标
scope	范围
skel	框架
standard	标准色
window list	窗口名列表
window name	窗口名
work area	工作区

第二章 FoxPro 命令

2.1 ? | ?? 运算

功能:

计算表达式的值并形成输出。

格式:

```
? | ?? [<expr1> [PICTURE <expC1>]
        [FUNCTION <expC2>]
        [AT <expN>]
        [, <expr2>] ...
```

?和??命令将计算表达式的值,并将结果送往屏幕、打印机或同时送往二者。

如果是 SET PRINTER ON,则这两个命令的输出将直接送往屏幕和打印机。但如果是 SET CONSOLE OFF,并且 SET PRINTER ON,则输出将只送往打印机。

?—— 单问号命令将在打出表达式结果之前先产生一个回车和换行符,除非使用了 FUNCTION 或 __ALIGNMENT 来专门说明,否则,它(?)都将在下一行的左边界处开始打印输出结果,如果没有指定任何表达式,则输出一个空行。

??—— 两个连续问号命令将从当前的位置处开始显示结果,在此之前不产生回车和换行符。

(1) PICTURE 子句

如果选用了 PICTURE 子句,则表达式的值将按 <expC1> 指定的格式进行显示。<expC1> 可由功能符、样本符或二者的组合所构成。

功能符影响表达式整体的格式,而样本符只对表达式中的单个字符施加影响。如果在 <expC1> 中用到功能符,则它们须出现在格式符的前面,并且由@ 先导。象所希望的那样,许多功能符可以立即跟在@ 后面,不必嵌入空格。其中最后一个功能符之后必须有一个以上的空格,空格标志了功能符的结束和样本符的开始。

(2) FUNCTION 子句

FUNCTION <expC2> 子句给 ? 和 ?? 命令的输出提供了包含功能符的可变手段。FUNCTION 子句的使用等价于在 PICTURE 子句中说明功能符,但在 FUNCTION 子句中不再需要@。

可用于 ? 和 ?? 命令的特定功能符有:

V<n>: 允许输出在限制水平位置数量下保持垂直方向的整齐。<n> 是用于

显示的最大列数。

另外的可用功能符和样本符将在@...SAY / GET 命令中列出。

(3) AT子句

AT子句用于指定输出显示的列数 <expN>。这个操作可用于将输出象表格一样地列出。

例:

```
? 15*(10+10)
```

```
? 'Welcome to'
```

```
?? 'FOXPRO'
```

参见: @... SAY / GET, SET MEMOWIDTH, SET PRINTER 命令和“系统内存变量”一章。

2.2 ??? 运算

功能:

将输出直接送往打印机而不改变函数PCOL()和PROW()的值。

格式:

```
??? <expC>
```

??? 命令将 <expC> 的内容直接送往打印机, 而不增加打印的列和行值, 这个命令还允许送打印控制符给打印机而不推进打印头。

打印机控制符允许重置打印机, 或者改变打印风格或大小, 设置粗体打印等。这些字符可由用户所使用的打印机的可打印字符和不可打印字符组成, 可以用下面几种不同方式来直接控制打印机:

- ① 使用 CHR() 函数直接送 ASCII 码给打印机;
- ② 用 { } 发送打印机控制字符或 ASCII 码值;
- ③ 使用 CHR() 函数和 { } 的组合来发送。

控制字符既可在打印开始之前, 也可在打印结束之后, 使用系统内存变量 __PSCODE 和 __PECODE 发往打印机。如想知道更详细的说明, 请参阅本手册中的“系统内存变量”一章。

不同打印机的控制字符是有差异的, 关于打印机控制字符请参考您的打印机的手册。

参见: ? / ??, CHR() 和“系统内存变量”一章。

2.3 @...SAY / GET 命令

功能:

格式输出或输入。

格式:

```
@ <row, col>
```

```
[SAY <expr1>
```

```

[PICTURE <expC1>]
[FUNCTION <fcodeS1>]]
[GET <VAR>
[PICTURE <expC2>]
[FUNCTION <fcodeS2>]
[RANGE [<expN1>][, <expN2>]]
[VALID <expL1> | <expN3>[ERROR <expC3>]]
[WHEN <expL2>]
[DEFAULT <expr2>]
    [MESSAGE <expC4>]]
[[OPEN] WINDOW <window name>]
[COLOR [<standard>][, <enhanced>]
 | COLOR SCHEME <expN4>]

```

使用@命令的各种形式，可以将输出按一定格式送往屏幕或窗口，或创建输入屏幕，或格式输往打印机。

(1) 屏幕 / 打印机座标 <row, col>

<row> 和 <col> 是数值型表达式，它决定输出出现的位置。对于屏幕 I/O，第一行的值为 0，最后一行是显示屏的高度减 1。一般来说最后一行是 24，除非是在扩展型的显示屏模式下运行。行编号一般从顶到底。对于打印机，实际页的大小决定了最大行数。

对于屏幕，首列编号为 0，最后一列是屏幕宽度减 1，一般情况下最后一列为 79，除非是在其它显示方式下运行。列编号从左到右。打印机的最大列数取决于实际的页宽。

(2) SAY 子句

如果出现 SAY 子句，则由 <expr1> 确定的值将从 <row, col> 位置开始显示，<expr1> 可以是一个 UDF (User-Defined Function 用户自定义函数)。如果选用了

```
SET DEVICE TO SCREEN
```

命令，则输出仅显示在屏幕上；如果选用命令

```
SET DEVICE TO PRINTER
```

则结果将直接输往打印机。@...SAY 子句也可以用在格式文件中。

● PICTURE 子句

如果选用 PICTURE 子句，则 <expr1> 的值将按说明符表达式 <expC1> 中指定的格式进行显示，这个字符串可由功能符 (Function Code)、PICTURE 样本符或二者的混合所组成。

如果用到了功能符，则在 <expC1> 中必须由 @ 作为先导首先出现，许多函数符可直接跟在 @ 之后，而不必嵌入空格。最后一个功能符之后必须有若干个空格字符，这些空格标志了功能符的结束和 PICTURE 样本符的开始。可

行的 FUNCTION 符和 PICTURE 样本符将在本章节的后面列出。

● FUNCTION 子句

FUNCTION 子句的 <fcodeS1> 给 @...SAY / GET 命令的输出提供了一个包括编辑功能符的可变手段。FUNCTION 子句的使用等价于 PICTURE 子句中的说明性功能符，但在 FUNCTION 子句中不再使用 @。

(3) GET 子句

如果选用了 GET 子句，则 <var> 的编辑将从 <row, col> 处开始，或者在 SAY 子句产生的输出后面的位置处开始。如果同时选用了 SAY 和 GET 子句，则在 SAY 的输出之后，GET 的结果之前自动插入一个字符。READ 命令必须与 GET 连用，才能编辑 GET 设定的域。SET DEVICE TO PRINTER 将自动忽略 GET。

● PICTURE <expC2> 及 FUNCTION <fcodeS2>

这两个子句与 SAY 中的目的是一样的，只不过这里用来编辑为响应 GET 而键入的数据。

● RANGE [<expN1>], <expN2>

RANGE 子句可用于对字符、日期和数值型变量指定输入值的范围。其值必须是字符、数值或日期表达式，这取决于 <var> 是否包含字符、数值或日期型数据。

如果键入的数据不在指定的范围内，则有一个信息会提示正确的范围。

RANGE 的上边界或下边界之一可以省略，但并不是两者都省略。如果省略了一个边界，则边界的这部分检查将跳过。如果不改变 <var> 的值，直接键入回车，也不执行对 RANGE 的边界检查。为了忽略省略边界的信息，请使用 ON READ ERROR 例程。

● VALID <expL1> | <expN3> [ERROR <expC3>]]

VALID 子句只允许合法的屏幕输入作为 @...GET 命令的一部分。当 VALID 子句与用户自定义的函数 (UDF) 连用时，可以大大简化屏幕输入的有效性检测。如果在 VALID 子句中调用一个 UDF，则 UDF 返回一个逻辑型或数值型的值。当试图离开 GET 域时，将用 <var> 的值与 VALID 的表达式进行比较。

注意：不同于 RANGE 的检查，VALID 的有效性检测总是执行的，除非是用 ESC 键离开。

① 使用逻辑表达式

如果 <expL1> 的值为真 (T.)，则输入被认为是正确的，READ 进入下一个字段域进行输入；如果 <expL1> 的值为假 (F.)，则键入值错误，FoxPro 将显示一个修改信息，直到按下空格后再进行数据输入为止。通常可用 ERROR 子句指定将错误信息送往某个地方。VALID 的合法性检查将继续到 <expL1> 为真或按下 ESC 键为止。

② 使用数值表达式

如果在 VALID 子句中用数值表达式取代逻辑表达式，则指令的执行取决