

IBM

IBMマルチステーション

5550

---

BASICインタープリター  
クック・ガイド

N: SX18-2200-1

IBM

IBMマルチステーション  
**5550**

BASICインタープリター  
クイック・ガイド

N: SX18-2200-1

## まえがき

このクイック・ガイドは、BASICのコマンド、ステートメント、関数、変数の構文集です。このクイック・ガイドの最初には、BASICを使うための基本事項が、最後には、エラー・メッセージの一覧が載せてあります。

## 目次

構文図の見方 .....	2
BASIC コマンド .....	3
変数型宣言文字 .....	3
演算子 .....	4
ファイル指定 .....	6
パス名 .....	6
装置名表 .....	6
1バイト・コード文字の連続コード .....	7
拡張コード .....	8
制御文字 .....	9
色 .....	10
キーボードと走査コード .....	11
コマンド、ステートメント、関数、変数 .....	13
メッセージ集 .....	83

第2版 1987年5月

このマニュアルは、製品の改良その他により適宜改訂されます。

© Copyright IBM Corp. 1986

## 構文図の見方

### 構文図の文字

大文字の英字	BASICの予約語。そのままの形で指定します。
小文字の英字 と日本語	定数, 変数名, ファイル名などに置きかえて指定します。 構文中では, ゴシック体で記述してあります。 次の小文字の英字は, 値の型を表しています。 x, y, z : 数式 n, m : 整数式 x\$, y\$ : スtring式 v : 数値変数 v\$ : スtring変数
その他	句読記号(ピリオド, コンマ, コロン, セミコロン, かっこ), ハイフン, 等号は, そのままの形で指定します。

### 構文図の読み方

- 線に沿って左から右に読む。
- 線の分岐点ではいずれか任意の線を選択する。
- 矢印の方向に沿って読み進む。

各パラメーターの機能は次のように表しています。

### コマンド, ステートメント



### 関数

— 関数名 (引数) —

### 変数

— 変数名 —

## BASIC コマンド

BASIC ファイル指定 <標準入力指定 > 標準出力指定 /I  
/F: ファイル数 /S: ブロック・サイズ /C: 通信バッファ  
/M: 最大作業領域: 最大ブロック・サイズ /D

ファイル指定 BASIC 始動後すぐに読み込んで実行するプログラムのファイル

<標準入力指定 標準入力になる装置またはファイル  
[省略時 KYBD:]

>標準出力指定 標準出力になる装置またはファイル  
(>>は指定ファイルに追加して出力する)  
[省略時 SCRNI:]

/I 選択肢/S: と/F: を有効にするための指定  
/F: ファイル数 プログラム実行中の一時点にオープンできるファイルの最大数 [最大 15] [省略値 3]

/S: ブロックサイズ ランダム・ファイルのバッファ・サイズ (バイト数) [省略値 128 バイト]

/C: 通信バッファ 非同期通信のデータ入力バッファ・サイズ (バイト数) [最大値 32767 バイト] [省略値 256 バイト]

/M: 最大作業空間, 最大ブロック 作業空間外に確保しておく領域のブロック・サイズ  
サイズ [ブロック単位 (1) ブロックは 16 バイト]

/D 数学関数を倍精度で計算する指定

注) 通常/I, /F, /S スイッチは指定する必要ありません。

## 変数型宣言文字

変数型 宣言文字	変数型	記憶空間 バイト数
\$	String	可変
%	整数	2
!	単精度数	4
#	倍精度数	8

## 演算子

### 算術演算子

#### 演算

べき乗  
負の記号  
乗算, 浮動小数点除算  
整数除算  
剰余の計算  
加算, 減算

#### 式の例

$X^Y$   
 $-X$   
 $X * Y, X / Y$   
 $X \div Y$   
 $X \text{ MOD } Y$   
 $X + Y, X - Y$

### 関係演算子

#### テストされる関係

等しい  
等しくない  
より小さい  
より大きい  
以下である  
以上である

#### 式の例

$X = Y$   
 $X <> Y, X > < Y$   
 $X < Y$   
 $X > Y$   
 $X < = Y, X < > Y$   
 $X > = Y, X = > Y$

=  
<>または><  
<  
>  
<=または=<  
>=または=>

### 論理演算子

#### NOT

X	NOT X
T	F
F	T

#### AND

X	Y	X AND Y
T	T	T
T	F	F
F	T	F
F	F	F

#### OR

X	Y	X OR Y
T	T	T
T	F	T
F	T	T
F	F	F

#### XOR

X	Y	X XOR Y
T	T	F
T	F	T
F	T	T
F	F	F

#### EQV

X	Y	X EQV Y
T	T	T
T	F	F
F	T	F
F	F	T

#### IMP

X	Y	X IMP Y
T	T	T
T	F	F
F	T	T
F	F	T

## ファイル指定

ファイル指定は、次の形式のストリング式で指定する。

装置名：ファイル名. エクステンション

装置名は、下記の装置名表のいずれかを指定する。

ファイル名は、1～8文字。エクステンションは1～3文字。

## パス名

パス名は、次の形式のストリング式で指定する。

装置名：パス名

パス名は、ディレクトリーを円記号(≡)で区切って並べたもので、63文字以内のストリング式で指定する。

## 装置名表

- KYBD:** キーボード。入力のみ。  
**SCRN:** 画面。出力のみ。  
**LPT1:** 第1プリンター。順次出力およびランダム出力。
- COM1:** 通信カード1。入力および出力。  
**COM2:** 通信カード2。入力および出力。
- A:** 第1ディスク・ドライブ。入力および出力。  
**B:** 第2ディスク・ドライブ。入力および出力。  
**C:** 第3ディスク・ドライブ。または、ハード・ディスク・ドライブ。入力および出力。

## 1バイトコード文字の連続コード

連続コード	文 字	連続コード	文 字	連続コード	文 字	連続コード	文 字
000	空白文字	030	→	060	<	090	Z
001	〒	031	←	061	=	091	[
002	ㇿ	032	(スペース)	062	>	092	\
003	ㇾ	033		063	?	093	]
004	ㇽ	034	-	064	@	094	^
005		035	#	065	A	095	_
006	=	036	\$	066	B	096	`
007		037	%	067	C	097	a
008		038	&	068	D	098	b
009		039	'	069	E	099	c
010		040	(	070	F	100	d
011		041	)	071	G	101	e
012		042	*	072	H	102	f
013		043	+	073	I	103	g
014		044	,	074	J	104	h
015		045	-	075	K	105	i
016	〒	046	.	076	L	106	j
017		047	/	077	M	107	k
018		048	0	078	N	108	l
019		049	1	079	O	109	m
020		050	2	080	P	110	n
021	〒	051	3	081	Q	111	o
022	〒	052	4	082	R	112	p
023	〒	053	5	083	S	113	q
024	〒	054	6	084	T	114	r
025	〒	055	7	085	U	115	s
026		056	8	086	V	116	t
027	」	057	9	087	W	117	u
028		058	:	088	X	118	v
029		059	:	089	Y	119	w

連続コード	文 字	連続コード	文 字	連続コード	文 字	連続コード	文 字
120	X	160	クベ-0	190	セ	220	ワ
121	Y	161	.	191	ン	221	.
122	Z	162	。	192	タ	222	.
123	—	163	、	193	チ	223	.
124	—	164	、	194	ツ	224	.
125	—	165	、	195	テ	.	.
126	—	166	ヲ	196	ト	.	.
127	—	167	フ	197	ナ	.	.
128	空白文字	168	イ	198	ニ	.	.
129	.	169	ウ	199	ヌ	.	.
.	.	170	エ	200	ネ	.	.
.	.	171	オ	201	ノ	.	.
.	.	172	カ	202	ハ	.	.
.	.	173	キ	203	ヒ	.	.
.	.	174	ク	204	フ	.	.
.	.	175	ケ	205	ヘ	.	.
.	.	176	コ	206	ホ	.	.
.	.	177	カ	207	マ	.	.
.	.	178	イ	208	ミ	.	.
.	.	179	ウ	209	ム	.	.
.	.	180	エ	210	メ	.	.
.	.	181	オ	211	モ	.	.
.	.	182	カ	212	ヤ	.	.
.	.	183	キ	213	ユ	.	.
.	.	184	ク	214	ヨ	.	.
.	.	185	ケ	215	ラ	.	.
.	.	186	コ	216	リ	252	.
.	.	187	サ	217	ル	253	.
.	.	188	シ	218	レ	254	.
159	.	189	ス	219	ロ	255	.

2バイト・コード文字の第1バイト



## 拡張コード

コード	キー操作
15	⏪ + ⏩
16-25	前面 + Q, W, E, R, T, Y, U, I, O, P
30-38	前面 + A, S, D, F, G, H, J, K, L
44-50	前面 + Z, X, C, V, B, N, M
59-68	PF1 ~ PF10
71	↵
72	↑
73	前面 + ↑
75	←
77	→
79	前面 + ↵
80	↓
81	前面 + ↓
82	挿入
83	削除
84-93	⏪ + PF1 ~ PF10
94-103	Ctrl + PF1 ~ PF10
104-113	前面 + PF1 ~ PF10
114	Ctrl + ⏪
115	Ctrl + ←
116	Ctrl + →
117	Ctrl + ↵
118	Ctrl + ↓
119	Clear
120-131	前面 + 1, 2, 3, 4, 5, 6, 7, 8, 9, 0, -, _
132	Ctrl + ↑
147	実行
152	取消
153	割込み
154	終了
156	前面 + ←
157	前面 + →

## 制御文字

連続コード	制御文字	名称
000	NUL	空白
001	SOH	ヘッディング開始
002	STX	テキスト開始
003	ETX	テキスト終結
004	EOT	伝送終了
005	ENQ	問合せ
006	ACK	肯定応答
007	BEL	ベル
008	BS	後退
009	HT	水平タブ
010	LF	改行
011	VT	垂直タブ
012	FF	書式送り
013	CR	復帰
014	SO	シフトアウト
015	SI	シフトイン
016	DLE	伝送制御拡張
017	DC1	装置制御 1
018	DC2	装置制御 2
019	DC3	装置制御 3
020	DC4	装置制御 4
021	NAK	否定応答
022	SYN	同期信号
023	ETB	伝送ブロック終結
024	CAN	取消
025	EM	媒体終端
026	SUB	置換キャラクタ
027	ESC	拡張
028	FS	ファイル分離キャラクタ
029	GS	グループ分離キャラクタ
030	RS	レコード分離キャラクタ
031	US	ユニット分離キャラクタ
127	DEL	抹消

# 色

カラー文字、4色カラー・グラフィック・モードのための16色

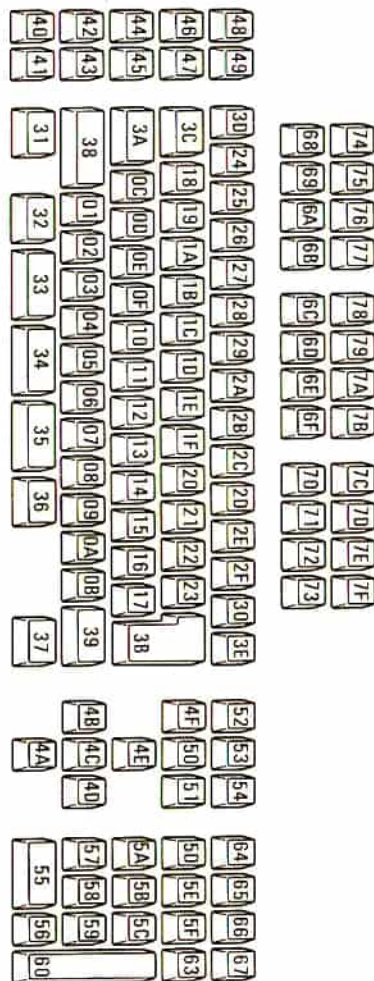
色番号	色	色番号	色
0	黒	8	灰色
1	青	9	薄い青
2	緑	10	薄い緑
3	水色	11	薄い水色
4	赤	12	薄い赤
5	紫	13	薄い紫
6	黄	14	薄い黄
7	白	15	明るい白

16色カラー・グラフィック・モードのための64色

	0	8	16	24	32	40	48	56
0	黒	暗い青	暗い緑	暗い水色	暗い赤	暗い紫	暗い黄	灰色
1	青	明るい青						薄い青
2	緑		明るい緑					薄い緑
3	水色			明るい水色				薄い水色
4	赤				明るい赤			薄い赤
5	紫					明るい紫		薄い紫
6	黄						明るい黄	薄い黄
7	白							明るい白

\* 図中の数字は色番号を表す。

# キーボードと走査コード





ABS 関数

絶対値を返す。

— ABS (x) —

x は, 変換する数値を指定する。

ASC 関数

最初の文字の連続コードを返す。

— ASC (x\$) —

x\$ は, 変換する文字列を指定する。

ATN 関数

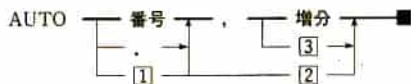
逆正接を返す。

— ATN (x) —

x は, 変換する数値を指定する。返る値は,  $-3.141593/2 \sim 3.141593/2$  のラジアン。

## AUTO コマンド

行番号を自動的に作成する。



- ① 10 (増分が指定されたときには0)
- ② 10
- ③ 前の値

番号には、最初の行番号を指定する。ピリオド(.)は、現在行を表す。

増分は、行番号の増分を指定する。

## BEEP ステートメント

スピーカーを鳴らす。

BEEP

## BSAVE コマンド

記憶域イメージのファイルを保管する。

BSAVE — ファイル指定, オフセット, 長さ

ファイル指定は、保管するファイルを指定する。  
オフセットは、DEF SEG で宣言したセグメントのオフセットを0～65535で指定する。  
長さは、保管する記憶域イメージのバイト数を1～65535で指定する。

## CALL ステートメント

機械語サブルーチンを呼び出す。

CALL — 数値変数 ( 変数 )

- ① 機械語サブルーチンに渡す引数は無し

数値変数は、サブルーチンの先頭記憶アドレスを示す数値変数名を指定する。変数は、機械語サブルーチンに渡す引数を指定する。

## BLOAD コマンド

記憶域イメージのファイルを読み込む。

BLOAD — ファイル指定, オフセット

- ① BSAVE で指定されたオフセット

ファイル指定は、読み込むファイルを指定する。  
オフセットは、DEF SEG で宣言したセグメントのオフセットを0～65535で指定する。

## CDBL 関数

倍精度に変換する。

— CDBL (x) —

x は、変換する数値を指定する。

## CDBL\$ 関数

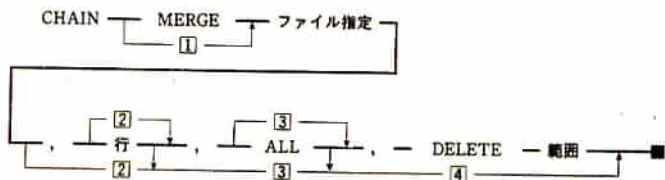
1バイト・コード英数字を2バイト・コード英数字に変換する。

— CDBL\$ (x\$) —

x\$ は、変換する1バイト・コードの英数字をstring式で指定する。

## CHAIN ステートメント

他のプログラムを読み込み、実行の制御を移す。



- ① マージしない
- ② 連鎖プログラムの最初の行
- ③ COMMON ステートメントで宣言した変数以外は渡らない
- ④ 現プログラムの行を削除しない

**MERGE** は、マージを指定。

**ファイル指定** は、制御を渡すプログラム・ファイル名を指定する。

**行** は、連鎖されたプログラムの実行開始行を指定する。

**ALL** は、全変数を渡すという指定。ALL を指定しない場合、変数を渡すためには現在記憶域にあるプログラム中に COMMON ステートメントが必要。

**DELETE** は、前回マージしたプログラムを削除するために、プログラム中の削除範囲を指定する。

## CHDIR コマンド

現行ディレクトリを変更する。

CHDIR — パス名 —

パス名は、変更するディレクトリのパス名を63文字以内のstring式で指定する。

## CHR\$ 関数

連続コードを文字に変換する。

— CHR\$ (  $\overbrace{\quad\quad}^{\downarrow}$  n  $\overbrace{\quad\quad}^{\downarrow}$  ) —

n は、文字に変換する連続コードを0~11535の範囲で指定する。

## CINT 関数

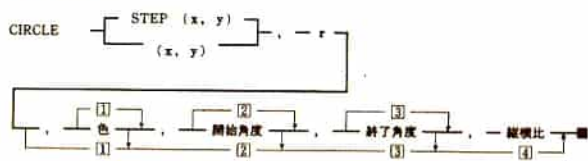
整数に変換する。

— CINT (x) —

x は、整数に変換する値を指定する。x の範囲は、-32768~32767。  
CINT は、x の小数点以下を四捨五入して、整数に変換する。

## CIRCLE ステートメント

楕円を描く。



- ① 前景色      ② 0  
③  $2 * 3.141593$       ④ 単色/16色カラーは1, 4色カラーは2

STEP(x,y) は、楕円の中心座標を相対形式で指定する。

(x,y) は、楕円の中心座標を絶対形式で指定する。

r は、楕円の長軸半径を指定する。

色は、楕円の色を次の範囲のパレット番号で指定する。省略時は、前景色。

画面モード	パレット番号
単色	0 / 1
4色カラー	0 ~ 3
16色カラー	0 ~ 15

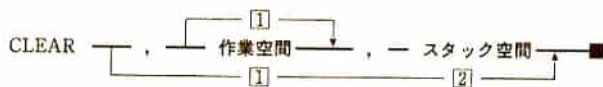
開始角度、終了角度は、楕円の始まりと終わりをラジアン ( $-2 * 3.141593$

$\sim 2 * 3.141593$ ) で指定する。省略時の開始角度は0, 終了角度は  $2 * 3.141593$

縦横比は、座標の縦横比を指定する。省略時は単色, 16色カラーのときに1, 4色カラーのときに2。

## CLEAR コマンド

記憶域のデータを消去する。



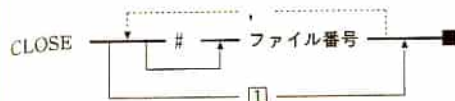
- ① 64K バイト  
② 使用可能な記憶域の1/8または512バイトのいずれか小さい方

作業空間は、確保しておく作業空間の最大バイト数を指定する。

スタック空間は、確保しておく作業空間のバイト数を指定する。

## CLOSE ステートメント

装置またはファイルをクローズする。

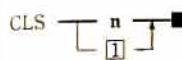


- ① オープンしているすべてのファイルと装置をクローズ

ファイル番号は、オープンしているファイルの番号を指定する。

## CLS ステートメント

画面を消去する。



- ① 下表を参照。

n は消去する範囲を0~2で指定する。

ビューポートが有るとき

CLS	画面
CLS 0	画面全体
CLS 1	ビューポート内のすべて
CLS 2	文字表示範囲のみ
CLS	ビューポート内のすべて
Clear	文字表示範囲のみ

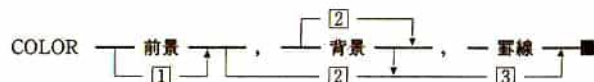
ビューポートが無いとき

CLS	画面
CLS 0	画面全体
CLS 1	ビューポート内のすべて
CLS 2	文字表示範囲のみ
CLS	ビューポート内のすべて
Clear	画面全体



## COLOR ステートメント (単色文字/単色グラフィック)

単色文字、単色グラフィック・モードのときに文字の下線づけ、反転、野線を指定する。



① ② ③ 前の値

前景は、次の重みで文字の下線づけ、高輝度、明滅を指定する。  
(単色グラフィック・モードは、下線づけのみ)

属性	明滅	高輝度	下線づけ
重み	16	8	1

背景は、文字の反転を0または1で指定する。

0: 反転なし      1: 反転

野線は、野線の種類を0~4で指定する。

0: 引かない, 1: 縦, 2: 横, 3: 縦と横,

4: 現在の画面の属性を変えない

## COLOR ステートメント (カラーグラフィック)

カラー・グラフィック・モードのときに背景色、標準パレット、野線を指定する。



① ② ③ 前の値

背景は、文字や図形の背景色を色番号0~15で指定する。

標準パレットは、次の標準パレットを0または1で指定する。

4色カラー・グラフィックのとき

	標準パレット0	標準パレット1
パレット	2 (緑)	3 (水色)
番号	4 (赤)	5 (紫)
番号	6 (黄)	7 (白)

16色カラー・グラフィックのとき

	標準パレット0	標準パレット1
1	36 (明るい赤)	9 (明るい青)
2	20 (茶色)	18 (明るい緑)
パレット	38 (やまぶき色)	27 (明るい水色)
番号	54 (明るい黄)	36 (明るい赤)
番号	34 (黄緑)	45 (明るい紫)
番号	26 (青緑)	54 (明るい黄)
番号	19 (水色)	63 (明るい白)
番号	3 (濃い水色)	56 (灰色)
番号	11 (緑青)	57 (薄い青)
番号	57 (薄い青)	58 (薄い緑)
番号	41 (すみれ色)	59 (薄い水色)
番号	45 (明るい紫)	60 (薄い赤)
番号	37 (濃いピンク)	61 (薄い紫)
番号	44 (赤紫)	62 (薄い黄)
番号	63 (明るい白)	63 (明るい白)

\*表中の番号は色番号

野線は、野線の種類を0~4で指定する。

0: 引かない, 1: 縦, 2: 横,

3: 縦と横, 4: 現在の属性を変えない

## COLOR ステートメント (カラー文字)

カラー文字モードのときに文字の色、野線の色、野線を指定する。



① ② ③ 前の値

前景は、文字の色を色番号0~15で指定する。文字を反転させるときは、色番号に16を加えた値を指定する。

野線色は、野線の色を色番号0~15で指定する。

野線は、野線の種類を0~4で指定する。

0: 引かない, 1: 縦, 2: 横, 3: 縦と横,

4: 現在の画面の属性を変えない

## COM(n)

通信割り込みを可能、禁止、保留にする。

COM (n) — ON —■  
COM (n) — OFF —■  
COM (n) — STOP —■

n は、通信カードを1または2で指定する。

1: 1枚目のカード, 2: 2枚目のカード



## COMMON ステートメント

連鎖されたプログラムに渡す変数名を指定する。

COMMON  変数 ■

変数は、連鎖されたプログラムに渡す変数名を指定する。配列変数には、変数名に"0"を付けること。

## CONT コマンド

プログラムの実行を再開する。

CONT —■

CONT は、**Break** キーを押した後、または STOP, END ステートメントの後、実行を再開させるが、中断中プログラムを編集すると再開できないので注意。

## COS 関数

余弦を返す。

— COS (x) —

x は、変換する数値を指定する。

## CSNG 関数

単精度に変換する。

— CSNG (x) —

x は、単精度に変換する数値

## CSNG\$ 関数

2 バイト・コード英数字を 1 バイト・コード英数字に変換する。

— CSNG\$ (x\$) —

x\$ は、変換する英数字をストリング式で指定する。

## CSRLIN 関数

現在のカーソルの行位置を返す。

— CSRLIN —

## CVI, CVS, CVD 関数

文字列型を数値型に変換する。

- CVI (2バイト・文字列) —
- CVS (4バイト・文字列) —
- CVD (8バイト・文字列) —

CVIは2バイト・文字列を整数に変換する。

CVSは4バイト・文字列を単精度数に変換する。

CVDは8バイト・文字列を倍精度数に変換する。

## DATA ステートメント

READ ステートメントで読み取る定数データを記憶する。

DATA  定数 ■

定数は、数値定数または文字列定数を指定する。

## DATA\$ 変数およびステートメント

日付の検索。

— DATE\$ —

日付を設定する。

DATE\$ =  x\$ ■

x\$ は、現在の日付を次の形式の文字列式で指定する。

mm-dd-yy

mm-dd-yy


mm-dd-yyyy

mm-dd-yyyy

(mm は月, dd は日, yy または yyyy は、年を表わす。)

## DEF FN ステートメント

ユーザー関数を定義する。

DEF FN  名前 ( 引数 ) = 式 ■

① 引数はなし

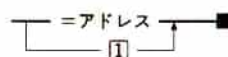
名前は、関数の名前となる変数名。

引数は、関数の引数となる変数名。

式は、関数が返すものを入れる変数名を指定する。式の型(数値または文字列)は、名前の型と一致していること。

## DEF SEG ステートメント

現在使用する記憶域内のセグメントを定義する。

DEF SEG  = アドレス ■

① BASIC のデータ・セグメントの先頭アドレス

アドレスは、セグメントの開始アドレスを0~65535で指定する。

## DEF 型ステートメント

変数の型を宣言する。

DEF 型  文字 文字 ■

型は、次の英字を指定する。

型	INT	SNG	DBL	STR
意味	整数	単精度数	倍精度数	文字列

文字は、型識別のための英字 A~Z を指定する。

## DEF USR ステートメント

機械語サブルーチンの開始アドレスを指定する。

DEF USR  $\overbrace{\quad\quad\quad}^n$  = オフセット  $\blacksquare$

□ 0

n は、サブルーチンの番号 0~9 を指定する。省略時は 0。

オフセットは、サブルーチンの開始アドレスを 0~65535 で指定する。

## DELETE コマンド

プログラム行を削除する。

DELETE  $\overbrace{\quad\quad\quad}^{\text{行番号 1}}$  -  $\overbrace{\quad\quad\quad}^{\text{行番号 2}}$   $\blacksquare$

行番号 1 は、最初の行番号を指定する。

行番号 2 は、最後の行番号を指定する。

ピリオド (.) は現在の行番号を表す。

## DIM ステートメント

配列変数を宣言する。

DIM  $\overbrace{\quad\quad\quad}^{\text{変数 (添字)}}$   $\blacksquare$

変数は、配列変数の名前。

添字は、配列の大きさと次元をカンマで区切った数値で定義する。

次元の最大値は、255。大きさを表す要素数の最大値は 32767、最小値は 0。

## DRAW ステートメント

線を描く。

DRAW— $\overbrace{\quad\quad\quad}^{\text{ストリング}}$   $\blacksquare$

ストリングは、次のコマンドをストリング式で指定する。

[コマンド]	[接頭コマンド]	
Un 上へ移動	En 右斜め上に移動	B 書かずに移動
Dn 下へ移動	Fn 右斜め下に移動	N 移動後、もとに戻る
Ln 左へ移動	Gn 左斜め下に移動	
Rn 右へ移動	Hn 左斜め上に移動	
Mx,y 相対的移動: x の前に+か-をつける		
絶対的移動: x の前につけない		

An 角度の設定 n=0: 0°, n=1: 90°, n=2: 180°, n=3: 270°

TAn 回転角度の設定 -360~360, +: 反時計回り, -: 時計回り

Cn パレット番号 省略時は前景色。

画面モード	パレット番号
単色	0 / 1
4色カラー	0 ~ 3
16色カラー	0 ~ 15

Sn スケール スケール因子は n/4 で、n の省略時は 4

X 変数 ストリング内で変数を使用するときに指定する。

P 絵の具、ふち 絵の具は図形の色をパレット番号 (C と同じ) で指定する。

ふちは図形のふちの色をパレット番号 (C と同じ) で指定する。

## EDIT コマンド

プログラム編集のために行を表示する。

EDIT  $\overbrace{\quad\quad\quad}^{\text{行番号}}$   $\blacksquare$

行番号は、表示する行番号を指定する。

ピリオド (.) は、行番号のかわりに現在行を表す。

## END ステートメント

プログラムの実行を終了する。

END —■

プログラムの実行を終了し、ファイルをすべてクローズしてコマンド・レベルに戻る。

## ENVIRON ステートメント

DOS のコマンド処理プログラムの環境に、パラメーターを追加したり修正を行なう。

ENVIRON — パラメーター指定 —■

パラメーター指定は、次の形のSTRING式で指定する。

名前 [=] [パラメーター]

名前には DOS のコマンド名やプログラム・ファイル名を指定する。

パラメーターには名前に対応するパラメーターを指定する。

名前とパラメーターは等号 (=) または、スペースで区切る。2 つ以上の等号 (=) またはスペースはパラメーターの一部とみなされる。パラメーターを環境から削除する場合は、パラメーターに空STRINGまたはセミコロン (;) を指定する。

## ENVIRON\$ 関数

DOS のコマンド処理プログラムの環境のパラメーターを検索する。

— ENVIRON\$ — ( — [名前  
n] — ) —

名前には、DOS のコマンド名やプログラム・ファイル名の一部分または全部をSTRING式で指定する。

n は、環境に設定されたパラメーターの順番を 1 ~ 255 の範囲で指定する。

## EOF 関数

ファイルの終了状態を返す。

— EOF (ファイル番号) —

ファイル番号は、オープンしたファイル番号を指定する。返る値は、ファイル終了時に -1 (真)、まだ終了していないときに 0 (偽)。

## ERASE ステートメント

配列を消去する。

ERASE — 配列名 —■

配列名は、消去する配列の名前を指定する。

## ERDEV および ERDEV\$ 変数

エラーが起きた装置名およびエラー情報を返す。

— ERDEV —

— ERDEV\$ —

ERDEV は、次の情報を持つ 2 バイトの数値を返す。

下位 8 ビット…DOS 割り込み INT24 (致命的エラー処理ルーチン・ベクトル) で返されるエラー・コードの値

上位 8 ビット…装置名ヘッダーの属性ビット (ビット 0 ~ 3, 13 ~ 15) の値

ERDEV\$ は、文字装置 (ディスプレイ、キーボード、プリンター) のときに 8 バイトの装置名を、ブロック装置 (ディスク) のときに 2 文字のドライブ指定子を返す。



## ERR および ERL 変数

エラーが起きたときに、その行番号とエラー・コードを返す。

— ERR —  
— ERL —

ERR は、エラー・メッセージ・コードを返す。  
ERL は、エラーの発生した行番号を返す。

## ERROR ステートメント

BASIC のエラーを模倣したりユーザ独自のエラーを発生させるために、エラーの状況を設定します。

ERROR — エラー・コード — ■

エラー・コードには、エラー発生と同時に ERR 変数に設定される値を、0～255 の範囲で指定する。

## EXTERR(n) 関数

DOS の拡張エラー情報を返す。

EXTERR — (n) — ■

**n** には求める情報を 0～3 で指定する。

- 0 : エラー・コード
- 1 : エラー・クラス
- 2 : 対処 (エラー処置のための指示)
- 3 : 場所 (エラーが起きた場所)

## EXP 関数

指数関数を計算する。

— EXP (x) —

**x** は、計算する数式を 88.02969 より小さい値で指定する。

## FIELD ステートメント

ランダム・ファイル・バッファー内に変数の領域を割り振る。

FIELD #          ファイル番号         , — 幅 — AS — スtring変数 — ■

ファイル番号は、オープンしたファイル番号を指定する。  
幅は、変数に割り振る領域の大きさをバイト数で指定する。  
String変数は、ランダム・ファイルのアクセスに使用するString型の変数名を指定する。

## FILES コマンド

ディレクトリー情報を表示する。

FILES          ファイル指定          ■

□ DOS 省略時ドライブの現行ディレクトリーにある全ファイルを表示

ファイル指定は、「装置名:ファイル名」または「ドライブ指定子:ファイル名」の形式のString式でファイルを指定する。



## FIX 関数

数値を切り捨てて整数に変換する。

— FIX (x) —

x は、変換する数式を指定する。

## FOR および NEXT ステートメント

一連の命令を繰り返して実行する。

FOR 変数 = x TO y STEP z

NEXT 変数

① STEP 1    ② 最も新しい FOR に対応する変数

変数は、繰り返す回数を持つ変数名を指定する。

初期値は、繰り返し一回目の値を指定する。

最終値は、繰り返しの最終回の値を指定する。

増分は、繰り返すたびに加算する値を指定する。

## FRE 関数

未使用の記憶域の大きさをバイト数で返す。

— FRE (仮引数) —

## GET ステートメント (ファイル)

ランダム・ファイルからランダム・ファイル・バッファへ、レコードを読み込む。また、通信ファイルから通信バッファへ読み込む。

GET # ファイル番号, レコード番号

① 次のレコード

ファイル番号は、オープンしたファイル番号を指定する。

レコード番号は、ランダム・ファイルのときにレコード番号を

1~4,294,967,295 の範囲で、通信ファイルのときに通信バッファから読み取るデータのバイト数と同じ範囲で指定する。

## GET ステートメント (グラフィック)

画面上の一定区域内の点を読み取る。

GET (x1, y1) - (x2, y2), 配列名

(x1,y1)-(x2,y2) は、読み取る範囲を絶対座標で指定する。

配列名は読み取った情報を入れる数値配列を指定する。

配列に必要なバイト数を求める式は次のとおり。

$4 + \text{INT}((X * n + 7) / 8) * Y$

X : 水平方向の点の数

Y : 垂直方向の点の数

n : 画素ビット (☞ 右図)

画面	n
単色	1
4色カラー	2
16色カラー	4

## GOSUB と RETURN ステートメント

サブルーチンに行き、戻る。

GOSUB 行

RETURN

行は、サブルーチンの開始行番号を指定する。

## GOTO ステートメント

指定した行番号へ分岐する。

GOTO 一行 ■

行は、分岐先の行番号を指定する。

## HEX\$ 関数

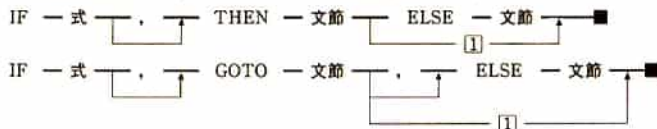
10進数値を16進数(文字列)に変換する。

— HEX\$ (n) —

nには、変換する数式を-32768~65535の範囲で指定する。

## IF ステートメント

式の結果に従って、プログラムの流れを判断する。



1 式が偽のときは何も実行せずに次のステートメントへ行く

式は、判断のための条件を数式で指定する。

文節は、1つのステートメントまたはコロン(:)で区切って並べた一連のステートメントを指定する。

式の値が真のときは、THENの後の文節のみが実行され、偽のときはELSEの後の文節のみが実行される。

## INKEY\$ 変数

キーボードから入力された最初の1文字を返す。

— INKEY\$ —

INKEY\$ は次の文字列が返る。

バッファ内に文字が無いとき…空文字列(長さ0)

1バイト・コード文字のとき…1文字文字列

2バイト・コード文字のとき…2文字文字列

## INP 関数

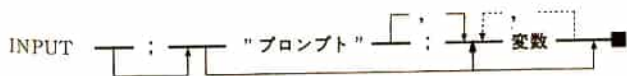
通信バッファ内の文字を1バイトだけ返す。

— INP (n) —

nには、通信バッファの開始アドレスを、0~65535の数値で指定する。

## INPUT ステートメント

プログラム実行中、キーボードからの入力を受け取る。



"プロンプト"はキーボードからの入力を要求するメッセージを文字列定数で指定する。

カンマ(,)とセミコロン(;)はプロンプトの後に疑問符(?)を表示するか否かの指定。

カンマ(,):疑問符(?)を表示しない。

セミコロン(;):疑問符(?)を表示する。

変数は、キーボードから入力された値を入れるために変数名を指定する。

## INPUT# ステートメント

順次データを読み込み、変数に割り当てる。

INPUT #ファイル番号, 変数

ファイル番号はオープンしたファイル番号を指定する。  
変数は、入力された値を入れるための変数名を指定する。

## INPUT\$ ステートメント

キーボードなどのファイルから読み取った文字を任意の数分返す。

INPUT\$ ( n, #ファイル番号 )

n は、ファイルから読み込む文字のバイト数を指定する。  
ファイル番号はオープンしたファイル番号を指定する。ファイル番号を省略すると、キーボードから入力したデータを返す。

## INPUT¥ ステートメント

キーボードなどのファイルから読み取った文字を任意の数分返す。

INPUT¥ ( n, #ファイル番号 )

n は、ファイルから読み込む文字のバイト数を指定する。  
ファイル番号は、オープンしたファイル番号を指定する。ファイル番号を省略すると、キーボードから入力したデータを返す。

## INSTR 関数

文字列を探索し、指定した文字の位置を返す。

INSTR ( n, x\$, y\$ )

例 1

n は、x\$ 中で y\$ 中の探索開始位置を先頭からのバイト数 1~255 で指定する。省略時は 1。

x\$ は、検索される文字列。

y\$ は、見つけ出す文字列。

## INT 関数

与えた数式の値以下で、最大の整数を返す。

INT ( x )

x は、整数に変換する数式を指定する。

## IOCTL ステートメント

装置駆動ルーチンに制御データを渡す。

IOCTL #ファイル番号, 制御データ

ファイル番号は装置駆動ルーチンのオープンに使用したファイル番号を指定する。

制御データには、装置駆動ルーチンに渡す制御データを、255文字以内のストリング式で指定する。制御データはコマンドをセミコロン (;) で区切って並べる。

## IOCTL\$ 関数

装置駆動ルーチンに渡した制御データの実行状態を確認する。

— IOCTL\$ ( # ファイル番号 ) —

ファイル番号は、装置駆動ルーチンのオープンに使用したファイル番号を指定する。

## JIS\$ 関数

ストリングの最初の文字の JIS 16 進コードをストリング形式で返す。

— JIS\$ (x\$) —

x\$ は、ストリング式を指定する。

## KEY ステートメント

PF キーの設定または表示を行う。

KEY — n, x\$ —■

任意の文字を表示・入力する機能を PF キーに割り当てる。

n は、割り当てる PF キーを 1~10, 30, 31 の整数式で指定する。  
x\$ は、割り当てる文字を 15 バイト以内のストリング式で指定する。

x\$ が CHR\$ (シフト)+CHR\$ (走査コード) のとき  
キー割り込みを行うための機能を割り当てる。

n は、キーを指す番号を 15~25 の整数式で指定する。  
シフトには、次のいずれかの 16 進コードを指定する。

CAPS ( [前面] + [英数] )	&H40
[前面]	&H08
Ctrl	&H04
[↑]	&H01, &H02, &H03 のいずれか

走査コードには、キーボードの走査コードを 16 進数の &H01 から &H7F (10 進数で 1~127) で指定する。☞ 「キーボードと走査コード」

KEY ON —■

KEY OFF —■

KEY LIST —■

PF キーに割り当てられている文字の表示を制御する。

KEY ON は、画面の 24 行目に PF キーの値の 6 バイトずつ表示する。

KEY OFF は、24 行目の表示を消す。

KEY LIST は、10 個の PF キーに割り当てられている文字を表示する。



## KEY(n)ステートメント

キー割り込みを可能、禁止、保留に設定する。

KEY (n) — ON —■

KEY (n) — OFF —■

KEY (n) — STOP —■

n は、割り込みを起こすキーの番号を 0 ~ 25, 30, 31 の範囲で指定する。

1 ~ 25 に対応するキー

番号	キー
1 ~ 10, 30, 31	PF1 ~ PF12
11	↑
12	←
13	↓
14	→
15 ~ 25	KEY で定義したキー

n に 0 を指定すると割り込み設定したすべてのキーを対象として制御できる。

KEY(n) ON は、割り込み処理を可能にする。

KEY(n) OFF は、割り込み処理を禁止にする。

KEY(n) STOP は、割り込み処理を保留にする。

## KILL コマンド

ディスク上のファイルを削除する。

KILL — ファイル指定 —■

ファイル指定には、「ドライブ指定子:ファイル名」形式のstring式で削除するファイルを指定する。省略時は、DOS の省略時ドライブが参照される。

## KLEN 関数

stringの文字数を返す。

— KLEN (x\$) —

x\$ には、string式を指定する。

## KPOS 関数

stringの n 番目の文字の位置をバイト数で返す。

— KPOS (x\$, n) —

x\$ には、string式を指定する。

n は、文字の位置を左からの文字数で指定する。x\$ の文字数より n が大きいと KPOS は、0 を返す。

## KTNS\$関数

stringの最初の文字の JIS 区点コードをstring形式で返す。

— KTNS\$ (x\$) —

x\$ には、string式を指定する。



## LEFT\$ 関数

文字列の左から n バイト分の文字を返す。

— LEFT\$ (x\$, n) —

x\$ は、文字列式を指定する。

n は、結果として返す長さを 0~255 のバイト数で返す。

## LEN 関数

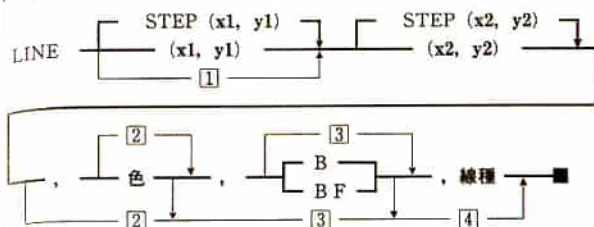
文字列の長さ (バイト数) を返す。

— LEN (x\$) —

x\$ は、文字列式を指定する。

## LINE ステートメント

画面上に直線または矩形を描く。



- ① 最終座標      ② 前景色  
③ 線を描く      ④ 実線

STEP(x1, y1) または (x1, y1) は、描き始めの座標値。  
STEP(x2, y2) または (x2, y2) は、描き終わりの座標値。  
(STEP 付きは相対座標, STEP なしは絶対座標)  
色は線の色をパレット番号で指定する。

画面モード	パレット番号
単色	0/1
4色カラー	0~3
16色カラー	0~15

B は矩形を描く指定。

BF は中塗り矩形の指定。

線は線種を &H0000~&HFFFF の 16 進数で指定する。

## LET ステートメント

式の値を変数に割り当てる。

LET 変数 = 式

変数は、式の値を受け取る変数または配列要素の名前を指定する。

式は、変数に割り当てる文字列式または数式を指定する。

## LINE INPUT ステートメント

キーボードから、行単位 (半角文字で 254 文字まで) の文字列を変数に読み込む。

LINE INPUT "プロンプト"; 文字列変数

"プロンプト" には、キーボードから入力要求するメッセージを文字列定数で指定する。

変数は、読み取った文字列を受け取る文字列変数または配列要素を指定する。

## LINE INPUT #ステートメント

順次ファイルから、行単位（半角文字 254 文字）のストリングを変数に読み込む。

LINE INPUT — #ファイル番号, ストリング変数 —■

ファイル番号は、順次ファイルのオープンに使用したファイル番号を指定する。

変数は、読み取ったストリングを受け取るストリング変数または配列要素を指定する。

## LIST コマンド

現在記憶域にあるプログラムを画面上または指定した装置に出力する。

LIST — [行1] — [行2] — , ファイル指定 —■

行番号 1 は出力する最初の行番号を、行番号 2 は出力する最後の行番号を 0 ~ 65529 の数式で指定する。

ピリオド (.) は現在行を表す。

ファイル指定は「装置名: ファイル名」の形式のストリング式で出力するファイルを指定する。ファイル指定を省略すると画面に出力される。

## LLIST コマンド

現在記憶域にあるプログラムをプリンターに出力する。

LLIST — [行1] — [行2] —■

行番号 1 は出力する最初の行番号を、行番号 2 は出力する最後の行番号を 0 ~ 65529 の数式で指定する。

ピリオド (.) は現在行を表す。

## LOAD コマンド

プログラムを指定した装置から記憶域に読み込む。

LOAD — ファイル指定 — , R —■

ファイル指定は、「装置名: ファイル名」または「ドライブ指定子: ファイル名」の形式のストリング式で読み込むファイルを指定する。

R は読み込んでファイルを引き続き実行する場合の指定。

## LOC 関数

ファイル中の現在の参照位置を返す。

— LOC (ファイル番号) —

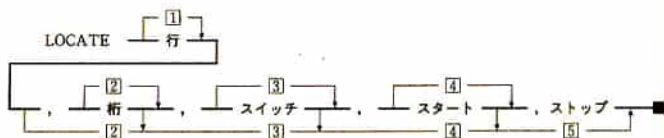
ファイル番号にはオープンしたファイル番号を指定する。

ランダム・ファイルの場合、LOC は最後に参照されたレコードのレコード番号を返す。

順次ファイルの場合は、現在までに読み込んだまたは書き込んだレコード数 (1 レコードは 128 バイト) を返す。

## LOCATE ステートメント

画面上のカーソル位置およびカーソルの大きさ、明滅を指定する。



- ① 現在の値
- ② 現在の値
- ③ 現在の値
- ④ 現在の値
- ⑤ スタートの値

行は、カーソル位置の行番号 1～24 を指定する。

桁は、カーソル位置の行番号 1～n を指定する。n は、画面に次のように依存する。

文字モード	.....80
単色/16色カラー・グラフィック・モード	.....78
4色カラー・グラフィック・モード	.....40

スイッチは、カーソルを表示するか否かを次の 0 または 1 で指定する。

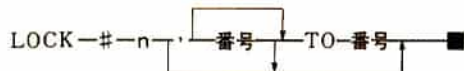
- 0 : カーソルを表示しない
- 1 : カーソルを表示する

スタートは、カーソルの開始走査線、ストップはカーソルの停止走査線を 0～n で指定する。n は、

16ドット・システム	
文字モード	.....20
グラフィック・モード	.....19
24ドット・システム	
文字/グラフィック・モード	.....28

## LOCK ステートメント

ファイルの全部または 1 部にアクセス禁止の指定をする。



n はオープンしたファイルの番号を指定する。

番号はアクセス禁止の範囲をレコード番号 (1～4,294,967,295) で指定する。

## LOF (ファイル番号)

ファイルの長さ (バイト数) を返す。

— LOF (ファイル番号) —

ファイル番号には、オープンしたファイル番号を使用する。

## LOG 関数

与えられた数式の自然対数 (底が e の対数) を返す。

— LOG (x) —

x は、1 以上の数式を指定する。

## LPOS 関数

プリンター・バッファ内のプリント・ヘッドの現在位置を返す。

— LPOS (n) —

n は、0 または 1 を指定する。(0 または 1 は、LPT1: の指定で 0 と 1 の違いはなし)

## LPRINT および LPRINT USING ステートメント

データをプリンターに出力する。

LPRINT 式のリスト ; 式のリスト ■  
①  
LPRINT USING v\$ ; 式のリスト ; 式のリスト ■

① 復帰/改行を行う

式のリストは、印刷する数式やストリング式をカンマ(,)かセミコロン(;)で区切って並べたものを指定する。

v\$ は印刷書式をストリングで指定する。印刷書式は PRINT USING ステートメント。

## MID\$ ステートメント

ストリングを置き換える。

MID\$ — (x\$, n, m) = y\$ ■  
①

① y\$の長さ

ストリング x\$ の n バイト目からを、ストリング y\$ の先頭から、長さ m バイト分で置き換える。

n は 1~255 の整数式。m は 0~255 の整数式。

## LSET および RSET ステートメント

ランダム・ファイル・バッファ内データに移行する。

LSET は左寄せ、RSET は右寄せで、ストリング変数によって識別されたフィールドに設定される。

LSET — ストリング変数=x\$ ■

RSET — ストリング変数=x\$ ■

ストリング変数は、FIELD ステートメントで定義した変数名を指定する。x\$ には、移行するデータをストリング式で指定する。

## MID\$ 関数

ストリング中の指定部分を返す。

— MID\$ — (x\$, n, m) —  
①

① n バイト目以降すべて

ストリング x\$ の n バイト目から、m バイトのストリングを返す。

n は 1~255 の整数式。m は 0~255 の整数式。

## MERGE コマンド

文字形式で保管されたプログラムを記憶域内にプログラムを重ね合わせる。

MERGE — ファイル指定 ■

ファイル指定は、ストリング式で、重ね合わせるファイルを指定する。

## MKDIR コマンド

ディスクに新規のディレクトリーを作成する。

MKDIR — パス名 ■

パス名は、63文字以内のストリング式でディレクトリーを指定する。



## MKI\$, MKS\$, MKD\$ 関数

数値型を文字列式に変換する。

— MKI\$ (整数式) —  
— MKS\$ (単精度数) —  
— MKD\$ (倍精度数式) —

MKI\$ は、整数を2バイト・文字列に変換。

MKS\$ は、単精度式を4バイト・文字列に変換。

MKD\$ は、倍精度式を8バイト・文字列に変換。

## NAME コマンド

ディスク・ファイルの名前を変更する。

NAME — 旧ファイル指定 — AS — 新ファイル名指定 —■

旧ファイル指定は名前を変更するファイルを、新ファイル指定は新しいファイルの名前を指定する。どちらも文字列式で指定する。

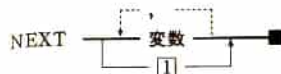
## NEW コマンド

記憶域にあるプログラムを削除し、すべての変数を消去する。

NEW —■

## NEXT ステートメント

FOR…NEXT ループを閉じる。



1 最も新しい FOR ステートメントに対応する

変数は、ループ変数を整数変数か単精度変数で指定する。省略時は一番近い NEXT と対応する。

## OCT\$ 関数

10進の数値を8進数で表す文字列に変換する。

— OCT\$ (n) —

n には、-32768~65535 の10進数の数式を指定する。

## ON COM(n) ステートメント

通信バッファに情報が入った時点で処理する通信割り込みルーチンを指定する。

ON COM (n) — GOSUB — 行 —■

n は、通信カードの番号を指定する。

行には、通信割り込みルーチンが始まる行番号を指定する。

行に0を指定すると、通信バッファに情報が入っても通信割り込みは起こらない。



## ON ERROR ステートメント

エラーが発生した場合に処理するエラー割り込みルーチンを指定する。

ON ERROR GOTO 一行 —■

行には、エラー割り込みルーチンが始まる行番号を指定する。行に0を指定すると、エラー割り込みは起こらず、エラー・メッセージを表示した後、実行が停止される。

## ON...GOSUB および ON...GOTO ステートメント

式の値に従って、複数の行番号の1つに分岐する。

ON — n — GOSUB 行 —■

ON — n — GOTO 行 —■

n は、0～255 の範囲の数式を指定する。

行は、分岐先の行番号をカンマ(,)で区切って指定する。

この構文は、nの値に従ってn番目の行番号に分岐する。

## ON KEY(n) ステートメント

指定したキーが押された場合に処理するキー割り込みルーチンを指定する。

ON KEY (n) — GOSUB 一行 —■

n は1～25, 30, 31の範囲の数式で割り込みを起こすキーを指定する。nの値は、次のキーと対応している。

n	キー
1～10, 30, 31	PF1 ~ PF12
11	↑
12	←
13	→
14	↓
15～25	KEY で指定したキー

行には、キー割り込みルーチンが始まる行番号を指定する。行に0を指定するとキーを押しても割り込みは起きない。

## ON PLAY(n) ステートメント

プログラム実行の背景で音楽演奏を継続するための、音楽割り込みルーチンを指定する。

ON PLAY (n) — GOSUB 一行 —■

ON PLAY(n)は、背景音楽バッファ内の音符の数がn以下の場合に割り込み処理ルーチンに行く。nは、1以上の音符数を指定する。nにバッファ内にある音符の数以上の値を指定すると割り込みは起きない。

行は、割り込みルーチンが始まる行番号を指定する。行に0を指定すると音符の数がn以下でも割り込みは起きない。

## ON TIMER(n) ステートメント

一定の時間が経過した時点で処理する割り込みルーチンを指定する。

ON TIMER (n) — GOSUB 一行 —■

n は、経過時間を秒で表す数式を1～86400の範囲で指定する。

行は、割り込みルーチンが始まる行番号を指定する。行に0を指定すると指定した時間が経過しても割り込みは起きない。

## OPEN ステートメント

ファイル, または装置への入出力ができるように初期設定 (オープン) する。

OPEN—ファイル指定—FOR—モード1—ACCESS—アクセス—  
 [1]  
 [2]  
 [3] ロック AS #—ファイル番号—LEN—レコード長—  
 [4]

- [1] RANDOM  
 [2] APPEND なら READ/WRITE, それ以外は WRITE  
 [3] LOCK READ WRITE  
 [4] 128

OPEN — モード 2, —

#—ファイル番号, ファイル指定—, レコード長—

ファイル指定はストリング式でオープンするファイルを指定する。  
 モード 1, モード 2 は, プログラム中のファイルのアクセス方法を下記のいずれかのモード名を使って指定する。(モード 1 はそのまま, モードは 2 重引用符付き)。

アクセス方式	モード 1	モード 2
ランダム入出力: レコード番号による入出力	RANDOM	R
順次出力: ファイルの先頭から順に出力	OUTPUT	O
順次入力: ファイルの先頭から順に入力	INPUT	I
順次出力 (付加): 既存レコードの次から順に出力	APPEND	A

アクセスはすでにオープンされている可能性のあるファイルに対してアクセスする場合に次のいずれかを指定する。

**READ** 読み取りモード

**WRITE** 書き込みモード

**READ WRITE** 読み取り/書き込みモード

ロックは, 他の処理からファイル・オープンの要求が行われた場合の対処を次のいずれかで指定する。

**SHARED** 読み書き可能

**LOCK READ** 読み取り不可

**LOCK WRITE** 書き込み不可

**LOCK READ WRITE** 読み書き不可

ファイル番号はファイルを参照するための番号を 1~255 の範囲の整数で指定する。

レコード長は, ランダム・ファイルのレコード長 (バイト数) を 1~32767 の範囲で指定する。省略時は 128 バイト。

## OPEN" COM...ステートメント

通信ファイルへの入出力ができるように初期設定 (オープン) する。

OPEN " COM — n — : 速度 [1] パリティ [2] データ [3] ストップ [4]

RS [5] CS [6] m1 [7] DS [8] m2 [9] CD [10] m3 [11] LF [12] PE [13]

" AS #—ファイル番号—LEN=バイト数—

- |                      |              |               |
|----------------------|--------------|---------------|
| 1) 300               | 2) E         | 3) 8          |
| 4) 2(110bps), 1(その他) | 5) RTSを制御しない | 6) CS1000     |
| 7) 0                 | 8) DS1000    | 9) 0          |
| 10) CDを制御しない         | 11) 0        | 12) 改行文字を送らない |
| 13) パリティ検査をしない       | 14) LEN=128  |               |

n は 1 か 2 で何枚目の ASYNC 通信カードを使用するかを指定します。

通信データに関する指定

パラメータ (上段: 指定する事柄, 下段: 指定できる値)

速度	ビット伝送速度をビット/秒 (bps) で指定する整数定数 110, 150, 300, 600, 1200, 2400, 4800, 9600 (省略値=300)
パリティ	送受信用のパリティを指定する 1 文字 S(SPACE): 送受信ともパリティ。ビット 0 として扱う O(ODD): 奇数パリティで送信と受信検査を行う E(EVEN): 偶数パリティで送信と受信検査を行う M(MARK): 送受信ともパリティ・ビット 1 として扱う N(NONE): 送受信ともパリティ無しとして扱う (省略値=E)
データ	データ・ビット数を示す整数定数 7, 8 (省略値=8)
ストップ	ストップ・ビットの数を指定する整数定数 1: 1ビット 2: 2ビット (省略値=伝送速度が 110 の場合は, 2, その他は 1)

## 通信制御に関する指定

パラメータ	パラメータを指定した場合の働き
RS	RTS (送信要求) 信号検査を制御する
CS(m1)	CTS (送信可能) 信号の検査を制御する (省略時…RS を指定したとき CS0, RS を指定しないとき CS1000)
DS(m2)	DSR (受信可能) 信号の検査を制御する (省略時…DS1000)
CD(m3)	CD (キャリア検出) 信号の検査を制御する (省略時…CD0)
LF	各復帰文字に続いて改行文字を送ります。
PE	パリティ検査を行う

注) m1, m2, m3 は、信号が送られるまでの許容時間を 0～65535 の範囲のミリ単位で指定する。

## OPTION BASE ステートメント

配列添字の最小値を宣言します。

OPTION BASE  $\overbrace{\quad n \quad} \blacksquare$

n は添字の最小値 0 または 1 を指定する。(OPTION BASE を指定しないときの最小値は 0)

## OUT ステートメント

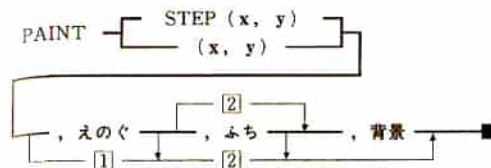
出力ポートへ 1 バイトのデータを送る。

OUT  $\overbrace{\quad n, m \quad} \blacksquare$

n は、0～65535 の範囲の数式で、ポート番号(I/O アドレス)を指定する。  
m は、データの連続コードを示す数式を、0～255 の範囲で指定する。

## PAINT ステートメント

画面上の区域を指定した色 (および模様) で塗りつぶす。



① ② 前景色

(x,y) と (x2,y2) は色を塗る区域内の点の座標を指定する。

STEP なし：絶対座標, STEP 付き：相対座標

えのぐには色または模様を指定する。色を塗るときはパレット番号を次の範囲で指定する。

画面モード	パレット番号
単色	0/1
4色カラー	0～3
16色カラー	0～15

模様を描くときは、「CHR\$(&Hnn)[+CHR\$(Hnn)]」…の形式で指定する。

ふちは、境界となる色を上記の範囲のパレット番号で指定する。

背景は、下地模様の中で境界線と区別する必要がある模様を CHR\$ 関数で指定する。



## PALETTE および PALETTE USING ステートメント

パレットに色を設定する。

PALETTE — パレット番号, 色 — ■

①

PALETTE USING — 配列名 — ■

4色カラー・グラフィック・モード		16色カラー・グラフィック・モード	
パレット番号	色番号 (色)	パレット番号	色番号 (色)
0	0 黒	0	0 黒
1	3 水色	1	9 明るい青
2	5 紫	2	18 明るい緑
3	7 白	3	27 明るい水色
		4	36 明るい赤
		5	45 明るい紫
		6	54 明るい黄
		7	63 明るい白
		8	56 灰色
		9	57 薄い青
		10	58 薄い緑
		11	59 薄い水色
		12	60 薄い赤
		13	61 薄い紫
		14	62 薄い黄
		15	63 明るい白

パレット番号は、パレットの区画の番号を次の範囲で指定する。

色は次の範囲の色番号でパレットに設定する色を指定する。

画面モード	パレット番号	色番号
4色カラー	0~3	0~15
16色カラー	0~15	0~15

配列名は色番号を配列要素を持った整数配列を指定する。

## PEEK 関数

記憶域から1バイト読み取る。

— PEEK (n) —

n は読み取る記憶域のアドレスを 0~65535 の範囲で指定する。

## PLAY ステートメント

音楽を演奏する。背景音楽の割り込み処理を許可、禁止、保留に設定する。

PLAY — スtring — ■

PLAY ON — ■

PLAY OFF — ■

PLAY STOP — ■

String は、次の音楽コマンドで構成されたものを指定する。

A~G に #, +, - をつけたもの A~G はラ~ソに対応。

On オクターブ。n は 0~6。省略時は 4。

>n 1 オクターブ上げる。n は音符。

<n 1 オクターブ下げる。n は音符。

Nn 音符 n を演奏します。n は 0~84。

Ln 音符の長さ 1/n。n は 1~64。

長さ	対応する音符
L1	全音符
L2	2分音符
L3	3連2分音符の1つ (4拍子の小節の1/3)
L4	4分音符
L5	5分音符 (1小節の1/5)
L6	3連4分音符の1つ
•	
•	
•	
L64	64分音符

Pn 休符。n は 0~64。L と同様に指定する。

• 付点音符。

Tn テンポ。1分間に演奏する4分音符の数を指定する。n は 32~255。省略時は 120。

MF 前景音楽。

MB 背景音楽。

MN ノーマル。各音符は L (長さ) で指定された時間の 7/8 の間演奏されます。

ML レガート。

MS スタッカート。

X変数: String変数を実行するための指定。



## PLAY 関数

背景音楽バッファにある音符の数を返す。

— PLAY (n) —

n は仮引数。

## PMAP 関数

ワールド座標系とスクリーン座標系の座標変換を行う。

— PMAP (z, n) —

z は、座標変換する点の座標値。

n は、返す値を 0～3 で指定する。

- 0 : ワールド座標系の x 座標値を  
スクリーン座標系の x 座標値に変換。
- 1 : ワールド座標系の y 座標値を  
スクリーン座標系の y 座標値に変換。
- 2 : スクリーン座標系の x 座標値を  
ワールド座標系の x 座標値に変換。
- 3 : スクリーン座標系の y 座標値を  
ワールド座標系の y 座標値に変換。

## POINT 関数 (点の色)

画面上の点の色 (パレット番号) を返す。

— POINT (x, y) —

(x,y) は調べる点の位置を絶対座標で指定する。  
与えられた座標が範囲外のときは -1 が返る。

## POINT 関数 (座標値)

現在の x, y 座標値を返す。

— POINT (n) —

n は、返す値を 0～3 で指定する。

- 0 : 現在のスクリーン座標系の x 座標値を返す。
- 1 : 現在のスクリーン座標系の y 座標値を返す。
- 2 : WINDOW が有効なとき、現在のワールド座標系の x 座標値を返す。  
WINDOW が無効なとき、現在のスクリーン座標系の x 座標値を返す。
- 3 : WINDOW が有効なとき、現在のワールド座標系の y 座標値を返す。  
WINDOW が無効なとき、現在のスクリーン座標系の y 座標値を返す。

## POKE ステートメント

記憶位置に 1 バイトのデータを書き込む。

POKE — n, m — ■

n はデータを書き込む記憶域のアドレスを 0～65535 の範囲で指定する。

m は、書き込むデータを 0～255 の範囲内で指定する。

## POS 関数

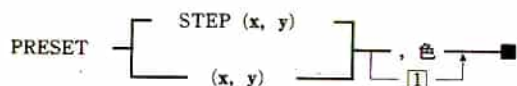
現在のカーソルの桁位置を返す。

— POS (n) —

n は仮引数です。

## PRESET ステートメント

画面上に点を描く。



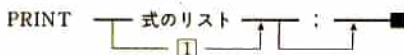
### 1 背景色

(x,y) は描く点の絶対座標で STEP (x,y) は相対座標で指定する。  
色は、点の色をパレット番号で指定する。省略時は背景色 (0)。

画面モード	パレット番号
単色	0/1
4色カラー	0~3
16色カラー	0~15

## PRINT ステートメント

画面上にデータを表示する。



### 1 復帰/改行を行う

式のリストは、表示する数値やストリングをカンマ、スペース、セミコロンのいずれかで区切って並べたもの。

各表示項目をカンマで区切ったときは、各項目が 14 文字単位の表示区域に表示される。

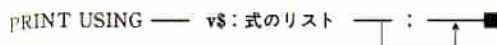
セミicolonまたは 1 つ以上のスペースで区切ったときは、各項目がつめて表示される。

; を式のリストの終わりに指定すると次の PRINT は同じ行から表示を始める。

; を省略すると復帰/改行が挿入される。

## PRINT USING ステートメント

データを指定した書式で画面に表示する。



v\$ は、書式指定文字 (下記参照) からなるストリング定数または変数。式のリストは、セミicolonやカンマで区切った表示データ (ストリング式、数式)。

### 書式指定文字

ストリングの表示

!	最初の文字のみ表示
& &	ストリングの左から「&間のスペース+2バイト」分を表示
@	入力どおりに表示

数値の表示

#	数値の桁の指定	¥¥	数の左に¥を表示
.	小数点位置の指定	,	3桁ごとに「,」を入れる
+	数値の符号を表示	---	指数書式の指定
-	負数のみ符号を表示	-	次の文字をリテラルとして表示
**	先行スペースを**で埋める		

## PRINT # ステートメント

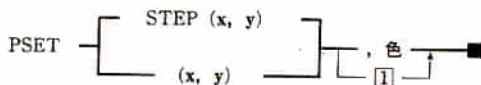
データを順番にファイルへ書き込む。



ファイル番号は、出力用にオープンしたファイル番号を指定する。  
v\$ は、書式指定文字 (PRINT USING) をストリング式で指定。  
式のリストは、ファイルに書き込むデータ (数値、ストリング式)。

## PSET ステートメント

画面上に点を描く。

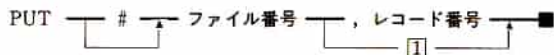


(x,y) は描く点を絶対座標で STEP(x,y) は相対座標で指定する。  
色は、点の色をパレット番号で指定する。省略時は前景色。

画面モード	パレット番号
単色	0/1
4色カラー	0～3
16色カラー	0～15

## PUT ステートメント (ファイル)

ランダム・ファイル・バッファからランダム・ファイルにレコードを書き込む。また通信バッファから通信ファイルヘデータを書き込む。

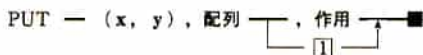


① 次のレコード番号

ファイル番号はオープンしたファイル番号を指定する。  
レコード番号は、ランダム・ファイルの場合には、書き込むレコード番号を、通信ファイルの場合、データのバイト数を、1～4,294,967,295 の範囲の数式で指定する。

## PUT ステートメント (グラフィック)

画面上の指定した領域に色を描く。



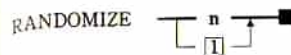
① XOR

(x,y) は、転送するイメージの左上隅の座標を使用する。  
配列は転送する情報を含む数値配列名を指定する。  
作用は、イメージの表示方法を表す、次のいずれかを指定する。

PSET  
PRESET  
XOR  
OR  
AND

## RANDOMIZE ステートメント

乱数シード (乱数発生のための初期値) を設定する。



① n を求めるプロンプトが出される

n は乱数シードとして用いる整数式を指定する。

## READ ステートメント

DATA ステートメントで定義した値を読み取る。



変数は読み取った値を割り当てる数値変数、ストリング変数、配列要素のいずれかを設定する。

## REM ステートメント

プログラムに注釈を入れる。

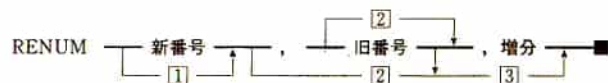


注釈には、プログラムの説明などの注釈をストリング (引用符なし) で指定する。



## RENUM コマンド

プログラムの行番号を付け替える。



- [1] 10
- [2] プログラムの最初の行
- [3] 10

新番号には、新しくつける行番号の最初の行番号を指定する。省略時は 10。  
旧番号には、付け替えを開始する現在のプログラムの行番号を指定する。  
省略時はプログラムの最初の行番号です。  
増分は新しくつける行番号の増分を指定する。

## RESET ステートメント

すべてのディスク・ファイルをクローズし、システム・バッファを消去する。

RESET —■

## RESTORE ステートメント

READ ステートメントで、再度 DATA ステートメントを読むために DATA ステートメントの行番号を指定する。

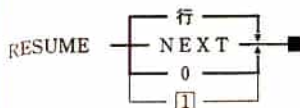


- [1] 最初の DATA ステートメントの行

行は、プログラムにある DATA ステートメントの行番号を指定する。  
省略時は、1 番最初の DATA ステートメントの行番号。

## RESUME ステートメント

エラー回復手順が行われた後、プログラムを再実行する。



- [1] 0

RESUME または RESUME 0 .....エラーを起こした行から再実行  
RESUME NEXT .....エラーを起こした次の行から再実行  
RESUME 行 .....行で指定した行番号から再実行

## RETURN ステートメント

サブルーチンから戻る。



- [1] 制御を受けた次の行

行は、プログラム中の戻る行番号を指定する。省略時は、割り込みの起きた場所に戻る。

## RIGHT\$ 関数

文字列 x\$ の右端から n バイト分の文字を返す。

— RIGHT (x\$, n) —

x\$ は文字列式を指定する。  
n は返す文字列の長さを整数式で指定する。



## RMDIR コマンド

ディスク上のディレクトリーを削除する。

RMDIR — バス名 — ■

バス名は削除するディレクトリーを 63 文字以内のストリング式で指定する。

## RND 関数

0 と 1 の間の乱数を返す。

— RND (x) — ■

x には、新規の乱数シードを設定するための数式を指定する。

x が負の場合……………新規の乱数を発生

0 の場合……………前回と同じ乱数を発生

正または省略時…乱数列において前回の次の乱数を発生

## RSET ステートメント

ランダム・ファイル・バッファー内にデータを移行する。

RSET — ストリング変数=x\$ — ■

ストリング変数は FIELD ステートメントで定義した変数名を指定する。

x\$ には、移行するデータをストリング式で指定する。

## RUN ステートメント

プログラムの実行を開始する。

RUN — 行 — ■  
①

① 最小の行番号

RUN — ファイル指定 — , R — ■  
②

② ファイルをすべてクローズする

行には、実行を開始するプログラムの行番号を指定する。

ファイル指定は、「装置名:ファイル名」または「ドライブ指定子:ファイル名」の形式のストリング式で実行するファイルを指定する。

R を指定すると、プログラムを記憶域に読み込むときに、現在オープン中のファイルはそのまま読み込む。

R を指定しないと、オープン中のファイルはクローズされる。

## SAVE ステートメント

記憶域にあるプログラムをディスクに保管する。

SAVE — ファイル指定 — [ , A ] [ , P ] — ■  
①

① 圧縮 2 進形式

ファイル指定には、ストリング式で保管するファイルを指定する。

A は、プログラムを文字形式 (内部コード形式) で保管する指定。

P は、プログラムをコード化 2 進形式で保管する。(この形式で保管されたプログラムは再度変更できないので注意)

A または P を省略するとプログラムは圧縮 2 進形式で保管される。

## SCREEN 関数

画面上の指定した行および列にある文字の連続コードを返す。

— SCREEN ( 行, 桁, z ) —

行は、1 から 24 の値を数式で指定する。

桁は、次の範囲の値を数式で指定する。

モード	桁の範囲
文字	1~80
単色/16色カラー・グラフィック	1~78
4色カラー・グラフィック	1~40

zには、連続コードを返すときに偽(0)を、色や属性を返すときに真(1)を指定する。

## SCREEN ステートメント

画面モードを設定する。

SCREEN モード , バースト

① ② 前の値

モードとバーストは次の組み合わせのいずれかで画面モードを設定する。

画面	単色	16ドット・カラー	24ドット・カラー
0,0	単色文字	単色文字	単色文字
0,1	単色文字	カラー文字	カラー文字
1,0	単色グラフ	単色グラフ	単色グラフ
1,1	単色グラフ	4色カラー・グラフ	16色カラー・グラフ
2,0	単色グラフ	単色グラフ	単色グラフ
2,1	単色グラフ	16色カラー・グラフ	16色カラー・グラフ

## SGN 関数

数式 x の符号を返す。

— SGN ( x ) —

x は符号を調べる数式を指定する。

## SHELL コマンド

DOS のコマンド・レベルで実行可能なコマンドを実行する。

SHELL — スtring — ■

String は、実行するコマンドをStringで指定する。

## SIN 関数

角度の正弦を返す。

— SIN ( x ) —

x は、ラジアン (度に 3.141593/180 を掛ける) で表した角度を数式で指定する。

## SOUND ステートメント

スピーカーから音を発生させる。

SOUND — 周波数, 持続時間 —■

周波数は、発生させる音を Hz (ヘルツ) で指定する。値の範囲は 37~32767。

持続時間は発生し続ける時間を 55 ミリ秒単位で指定する。

値の範囲は 0~65535。

音名と周波数

音名	周波数	音名	周波数
C	130,810	C	523,250
D	146,830	D	587,330
E	164,810	E	659,260
F	174,610	F	698,460
G	196,000	G	783,990
A	220,000	A	880,000
B	246,940	B	987,770
C*	261,630	C	1046,500
D	293,660	D	1174,770
E	329,630	E	1318,500
F	349,230	F	1396,900
G	392,000	G	1568,000
A	440,000	A	1760,000
B	493,880	B	1975,500

\* 中央の C

## SPACE\$ 関数

n 個のスペースをストリングで返す。

— SPACE\$ (n) —

n はスペースの数を 0~255 の範囲で指定する。

## SPC 関数

n 個のスペースを置く。

— SPC (n) —

n はスペースの数を 0~255 の範囲で指定する。

SPC は、PRINT, PRINT USING, LPRINT, LPRINT USING, PRINT #, PRINT # USING の中でのみ使用可能。

## SQR 関数

平方根を返す。

— SQR (x) —

x は 0 以上の数式を指定する。

## STOP ステートメント

プログラムの実行を停止し、コマンド・レベルに戻る。

STOP —■

## STR\$ 関数

数値型を文字列型に変換する。

— STR\$ (x) —

x は、文字列に変換する数式を指定する。

変換の対象となるものは、連続コードの 43(+), 45(-), 46(.) と数字に対応する 48~57。

## STRING\$ 関数

連続コードの m の文字を n 個並べた文字列、または x\$ の最初の文字を n 個並べた文字列を返す。

— STRING\$ (n,  $\left[ \begin{array}{c} x\$ \\ m \end{array} \right]$ ) —

n は、並べる文字の数を 1 バイト・コードの場合は 0~255, 2 バイト・コード文字の場合は 0~127 の範囲で指定する。

m は、文字の連続コードを指定する。

x\$ は、並べる文字の文字列を指定する。

## SWAP ステートメント

2 つの変数の値を交換する。

SWAP — 変数 1, 変数 2 — ■

変数 1, 変数 2 は交換する変数または配列要素を指定する。

2 つの変数は同じ型であること。

## SYSTEM コマンド

BASIC から DOS に戻る。

SYSTEM — ■

## TAB 関数

行の先頭から (n-1) 桁目まで、表示する文字のない桁に、スペースを置く。

— TAB (n) —

n は文字を表示する位置を 1~255 の範囲で指定する。

TAB は、PRINT, PRINT USING, LPRINT, LPRINT USING, PRINT #, PRINT # USING の中でのみ使用可能。

## TAN 関数

角度の正接を返す。

— TAN (x) —

x はラジアン (度に 3.141593/180 を掛けたもの) で表した角度を数式で指定する。



## TIMES\$ 変数

現在の時刻を返す。

— TIMES\$ —

hh:mm:ss の形式の 8 文字ストリングが返る。

## TIMES\$ ステートメント

時刻を設定する。

TIMES\$ = — x\$ —■

x\$ は時刻 hh:mm:ss の形式で指定する。

hh には時間を 0~23 の範囲で指定。省略不可。

mm には分を 0~59 の範囲で指定。省略時は 00 が設定される。

ss は指定しても常に 00 が設定される。

## TIMER 関数およびステートメント

現在の時刻を単精度の秒単位で返す。

経過時間 (秒) ごとの割り込み処理を可能、禁止、保留に設定する。

— TIMER —

TIMER — ON —■

TIMER — OFF —■

TIMER — STOP —■

TIMER ON は、割り込み処理を可能にする。

TIMER OFF は、割り込み処理を禁止にする。

TIMER STOP は、割り込み処理を保留にする。

## TRON および TROFF コマンド

プログラムの実行状態をトレースして調べる。

TRON —■

TROFF —■

TRON はトレース開始。

TROFF トレース終了。

## UNLOCK ステートメント

アクセス禁止を解除する指定を行う。

UNLOCK—#—n — 番号 — TO—番号 —■

n はオープンしたファイルの番号を指定する。

番号はアクセス禁止解除の範囲をレコード番号 (1~4,294,967,295) で指定する。

## USR 関数

機械語サブルーチン呼び出す。

— USR — x — (引数) —  
          └─┬─┘  
          └─┬─┘  
          ①

n は、DEF USR で機械語サブルーチンに指定した 0~9 の番号を指定する。

引数は数式またはストリング変数を指定する。

## VAL 関数

文字列  $x\$$  の数値を返す。

— VAL( $x\$$ ) —

$x\$$  は文字列式。

## VARPTR 関数

変数またはファイル制御ブロックのアドレスを返す。

— VARPTR ( —  $\left[ \begin{array}{c} \text{変数} \\ \text{\#ファイル番号} \end{array} \right]$  ) —

変数は、アドレスを調べるプログラム中の変数。  
ファイル番号は、ファイルをオープンしたときに指定した番号。返されるアドレスは、0～&HFFFF (65535) の範囲。

## VARPTR\$ 関数

変数のアドレスを文字列で返す。

— VARPTR\$ (変数) —

変数はアドレスを調べるプログラム中の変数を指定する。  
VARPTR\$ は次の3バイト・文字列を返す。

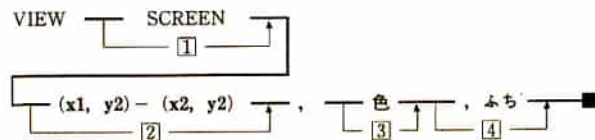
バイト0	バイト1	バイト2
型	アドレスの低位バイト	アドレスの高位バイト

型は次の値で変数型を示す。

2：整数，3：文字列，4：単精度数，8：倍精度数

## VIEW ステートメント

スクリーン座標系におけるビューポートを指定する。



- ① 相対座標                      ② スクリーン座標系  
③ 背景色                        ④ 境界線を描かない

SCREEN は指定するとスクリーン座標系に対して絶対座標に、省略すると相対座標になる。

( $x1,y1$ )-( $x2,y2$ ) はビューポートの領域を示す座標系になる。  
色はビューポートに塗るパレット番号を次の範囲で指定する。

画面モード	パレット番号
単色	0/1
4色カラー	0～3
16色カラー	0～15

\*色を省略するとビューポートは塗りつぶされない。

ふちは、ビューポートの境界線を描く色を上記と同様のパレット番号で指定する。

## VIEW PRINT ステートメント

文字の行方向表示範囲を設定する。

VIEW PRINT 行1 TO 行2 ■  
①

- ① 行1～行24の範囲

行1は、表示範囲の最初の行番号を1～24の範囲で指定する。  
行2は、表示範囲の最後の行番号を1～24の範囲で指定する。

## WAIT ステートメント

装置の入力ポートの状態をモニターする間、プログラムの実行を保留する。

WAIT — ポート, n —, m — ■  
                  └──┬──┘  
                  ①

① 0

ポートには、0～65535の範囲のポート番号を指定する。

n と m には、入力ポートのビットを調べるために使用する。n は調べる位置を、m は調べるビットのパターンを指定する。n と m の範囲は 0～255。m で指定するビット・パターンは、調べるビットが 0 なら 1 を、1 なら 0 を指定する。

## WHILE および WEND ステートメント

与えられた条件が真である間、ループ中の一連のステートメントを実行する。

WHILE — 式 — ■  
WEND — ■

式は条件となる数式を指定する。

## WIDTH ステートメント

装置への出力幅を設定する。

WIDTH — 桁 —, 行 — ■  
          └──┬──┘  
          ① ②

① ② 前の値

画面の表示範囲を設定する。桁と行の範囲は次のとおり。

桁 1～80 (4色カラーのときは1～40)

行 1～24

WIDTH — 装置, 幅 — ■

この WIDTH が実行された後、オープンされた装置に有効な幅割り当てを行う。

装置には、SCRN:,LPT1:,COM2: のいずれかを指定する。

幅は、装置に出力する行の幅を 0～255 の範囲で指定する。

WIDTH — ファイル番号, 幅 — ■

すでにオープンされている装置への出力幅を割り当てる。

ファイル番号は、装置をオープンしたファイル番号を 0～15 の範囲で指定する。

幅は、装置に出力する行の幅を 0～255 の範囲で指定する。



## WINDOW ステートメント

ワールド座標系におけるウィンドウを定義する。

WINDOW    SCREEN    (x1, y1) - (x2, y2) —■  
                  └──┬──┘  
                  ①

### ① 直交座標系

SCREEN は、指定すると x 座標値は左から右に、y 座標値は上から下へ増加する。省略すると x 座標値は左から右に、y 座標値は下から上へ増加する。

(x1,y1)-(x2,y2)はウィンドウの範囲を指定する。

## WRITE ステートメント

データを画面に表示する。

WRITE    式のリスト    ■  
          └──┬──┘  
          ①

### ① 復帰/改行を行う

式のリストは、表示する数式やストリング式、またはその両方のリストを記述する。

各式は、カンマまたはセミコロンで区切る。

WRITE は、出力項目が数値のときにカンマを、ストリングのときに引用符をつけて出力する。

## WRITE # ステートメント

データを順次ファイルに書き込む。

WRITE # ファイル番号, 式のリスト —■

### ① 復帰/改行を行う

ファイル番号は、出力用にオープンしたファイルの番号を指定する。

式のリストは、表示する数式やストリング式、またはその両方のリストを記述する。

WRITE # は、出力項目が数値のときにカンマを、ストリングのときに引用符をつけて出力する。

## メッセージ集

### 番号    メッセージ

- 54 **Bad file mode** (ファイル・モードの誤り)  
順次ファイルまたはクローズされたファイルに対して、PUT や GET を使おうとした。または、入力、出力、アペンド、またはランダム以外のファイル・モードで OPEN を実行した。  
文字形式 (内部コード形式) でないファイルをマージしようとした。
- 64 **Bad file name** (ファイル名の誤り)  
BLOAD, BSAVE, KILL, NAME, OPEN, または FILES ステートメントのファイル名に、正しくない形式が使われている。
- 52 **Bad file number** (ファイル番号の誤り)  
ファイル番号の指定が誤っている。
- 63 **Bad record number** (レコード番号の誤り)  
PUT または GET ステートメント中で、レコード番号が許された最大値 4,294,967,295 より大きいかゼロである。
- 17 **Can't continue** (継続不可能)  
次のようなプログラムを続行するため CONT を使おうとした。
- エラーのために停止したプログラム。
  - 実行の中断の間に修正したプログラム。
  - 存在しないプログラム。
- \*\* **Can't continue after SHELL** (SHELL ステートメント後の実行ができない)  
SHELL ステートメントによる子プログラムの実行後、メモリーが足りず BASIC を続けられない。(エラー番号なし)
- 69 **Communication buffer overflow** (通信バッファのあふれ)  
通信入力ステートメントが実行されたが、入力バッファがすでにいっぱいになっている。
- 25 **Device Fault** (装置の故障)  
インターフェース・アダプターからハードウェア・エラー表示が返された。  
OPEN"COM... ステートメントでテストされた信号が指定された時間内に検出されなかった。
- 57 **Device I/O Error** (装置入出力エラー)  
装置入出力操作中にエラーが起こった、DOS はエラーから回復できない。



- 24 **Device Timeout (装置タイムアウト)**  
BASIC が予定の時間内に入出力装置から情報を受け取らなかった。  
OPEN"COM... でテストされた各信号が、指定された時間内に検出されなかった。
- もう一度操作してみてください。
- 68 **Device Unavailable (操作が使用不可能)**  
存在しない装置に対してファイルをオープンしようとした。
- 66 **Direct statement in file (ファイル中に直接ステートメントがある)**  
文字形式 (内部コード形式) ファイルの読み込み、または連鎖中に、直接ステートメントに出会った。
- 61 **Disk full (ディスクがいっぱいである。)**  
ディスク記憶空間がすべて使用されている。
- 72 **Disk Media Error (ディスク・メディア・エラー)**  
アダプター・カードが、ハードウェアまたはメディアの故障を感知した。ディスクが悪くなっている。
- 71 **Disk not Ready (ディスクが作動可能でない)**  
ディスク・ドライブのレバーが開いている、あるいはディスク・ドライブに入っていない。
- 11 **Division by zero (ゼロでの除算)**  
式の中で、ゼロで除算をしようとした、またはゼロに負の数の異乗を行おうとした。
- 10 **Duplicate Definition (二重定義)**  
同じ配列に対してサイズを2回定義しようとした。これは次のような場合が考えられる。
- 2つのDIMステートメントで同じ配列を定義した。
  - 省略時の10の次元が与えられたあと、プログラムはその配列を定義するDIMステートメントに出会った。
  - 配列がDIMステートメントまたは省略時値で定義された後、プログラムはOPTION BASEステートメントに出会った。
- 50 **FIELD overflow (FIELDのあふれ)**  
ランダム・ファイルのレコード長としてOPENステートメント内で指定したバイト数より多いバイト数をFIELDステートメントが割り振ろうとしている。または、ランダム・ファイルへの順次入出力 (PRINT #, WRITE #, INPUT #) 中にFIELDバッファの終わりが起こった。
- 58 **File already exists (ファイルがすでに存在する)**  
NAMEステートメントで指定されたファイル名が、ディスク上

ですでに使用されている。

- 55 **File already open (ファイルはすでにオープンされている)**  
順次出力またはアペンド用にファイルをオープンしようとしたが、ファイルはすでにオープンされていた。また、オープンしているファイルに対してKILLを行おうとした。
- 53 **File not found (ファイルが見つからない)**  
LOAD, KILL, NAME, FILES, またはOPENステートメントで参照しているファイルが、指定したドライブ内のディスク上に存在しない。
- 26 **FOR without NEXT (NEXTのないFOR)**  
FORステートメントに対応するNEXTがない。
- 12 **Illegal direct (不正な直接モード)**  
直接モードでは使えないステートメント (DEF FN など) を直接モードで入力しようとした。
- 5 **Illigal function call (違法関数呼び出し)**  
有効範囲外のパラメーターがシステム関数に渡された。
- また、次のような場合の結果として起きた。
- 負または不当に大きな添字
  - 負の数に整数でない数の異乗を行おうとした。
  - DEFUSRが開始アドレスを定義しないうちに、USR関数を呼び出した。
  - GET (ファイル) またはPUT (ファイル) ステートメント上の負のレコード番号。
  - 関数またはステートメントの引数が不適切。
  - 記憶保護されたBASICプログラムにリストまたは編集を行おうとした。
  - DELETEステートメントによって存在しない行番号を削除しようとした。
- 52 **Input past end (入力が終わりを越えた)**  
空白 (空の) ファイルに対し入力ステートメントが実行された、または順次ファイル中の全データがすでに入力された後に、入力ステートメントが実行された。

出力またはアペンド用にオープンされたファイルから読もうとした場合にも起こる。

51 **Internal error (内部エラー)**

BASICで内部エラーが起こった。

23 **Line buffer overflow (行バッファのあふれ)**

文字の多すぎる行を入力しようとした。

22 **Missing operand (オペランドがない)**

式の中に、後ろにオペランドのない演算子、\*やORなどがある。

1 **NEXT without FOR (FORのないNEXT)**

NEXTステートメントに対応するFORステートメントがない。

19 **No RESUME (RESUMEがない)**

エラー状態またはERRORステートメントの結果、活動中のエラー割り込み処理ルーチンにプログラムが分岐したが、そのルーチンにRESUMEステートメントがなかった。(エラー割り込み処理ルーチン内で、物理的なプログラムの終わりに出会った。)

4 **Out of data (データがない)**

READステートメントが、DATAステートメントで与えられた以上のデータを読もうとしている。

7 **Out of memory (メモリーを越えた)**

プログラムが大きすぎる。FORループまたはGOSUBが多すぎる、変数が多すぎる、式が複雑すぎる、あるいはPAINTステートメントの指示が複雑すぎる。

14 **Out of string space (ストリング空間が足りない)**

BASICは、記憶域がいっぱいになるまでストリング空間を自動的に割り振るため、ストリング変数によりBASICがハウス・クリーニング後残っていた自由記憶域容量を越えたことを示す。

6 **Overflow (桁あふれ)**

数の絶対値が大きすぎて、BASIC数値形式では表現できない。

75 **Path/file access error (パス指定の誤り)**

MKDIR, CHDIR, RMDIRまたは他のファイル指定におけるパス名の指定で、DOSが、ファイルへのパスの指定に誤りを見つけた。

76 **Path not found (パスが見つからない)**

MKDIR, CHDIR, RMDIRまたは他のファイル指定におけるパス名の指定で、指定されたパスをDOSが見つけれなかった。

70 **Permission denied (書き込み要求拒否)**

このエラーは次の状態の時に起こります。

- 書き込み防止されているディスクに対して書き込みを行おうとした。
- ファイルをオープンする際にOPENステートメントの指定に誤りがあった。

- すでにOPENステートメントまたはLOCK, UNLOCKステートメントで書き込み防止についての処置がなされているファイルに対して再び設定しようとした場合。

- 他の処理によってすでに書き込み保護されているファイルに対して書き込みを行おうとした。

- データの書き込み終了後、他の処理がそのファイルを書き込み保護してアクセスしている時に、ファイルをクローズしようとした。

他の処理が書き込み保護の解除を行うまでファイル・クローズの命令を繰り返してください。

このメッセージは、ENDステートメントで同様の処理を行っている場合にも起こります。

- LOCK, UNLOCKステートメントを使用する前にDOSのSHAREコマンドが実行されていなかった。

74 **Rename across disks (ディスクを超えて指定した)**

ファイル名を新たに変更するときに現行のディスクではなく他のディスクを指定した。

20 **RESUME without error (エラーなしのRESUME)**

エラーを割り込み処理していないのにプログラム中にRESUMEステートメントがあった。

3 **RETURN without GOSUB (GOSUBのないRETURN)**

RETURNステートメントより前に、対応するGOSUBステートメントがない。

16 **String formula too complex (ストリング式が複雑すぎる)**

ストリング式が長すぎるか複雑すぎる。

15 **String too long (ストリングが長すぎる)**

255文字より長いストリングを作ろうとした。

9 **Subscript out of range (添字が範囲外)**

配列要素の添字が配列の次元を越えているか、あるいは間違っている。

2 **Syntax error (構文エラー)**

プログラム行の中に、対になっていないかっこや、綴りの間違ったコマンドやステートメントのような誤った文字列、または誤った句読点があるか、DATAステートメント内のデータがREADステートメント内の変数タイプ(数値またはストリング)と一致しない。

67 **Too many files (ファイルが多すぎる)**

現行ディレクトリー項目がすべていっぱいするとき、またはファイル指定が誤っているときに、新しいファイルを作成しようとした。

(SAVE または OPEN を使って)。

**13 Type mismatch (タイプが合わない)**

数値を必要とする場所にストリング値を与えた、またはストリング値のあるべき所に数値を入れた。

単精度数と倍精度数のように違った型の変数を SWAP しようとした。

**8 Undefined line number (行番号が存在しない)**

ステートメントまたはコマンド内で参照している行番号が、プログラム中に存在しない。

**18 Undefined user function (ユーザー関数が未定義)**

DEF FN ステートメントでまだ定義されていない関数を呼び出した。

**-- Unprintable error (印刷不可能エラー)**

発生したエラー状態に対して該当するエラー・メッセージがない。  
ERROR ステートメントでエラー・コードを定義していない。

**30 WEND without WHILE (WHILE のない WEND)**

対応する WHILE が実行されないうちに、WEND が現れた。

**29 WHILE without WEND (WEND のない WHILE)**

WHILE ステートメントに対応する WEND がない。すなわち、プログラムの物理的終わりになったとき WHILE がまだ活動状態だった。

**\*\* You can not SHELL to BASIC (子プログラムとしての BASIC を実行しようとした)**

子プログラムとしての BASIC の実行が発見された。(エラー番号なし)



N: SX18-2200-1

IBM プリチステーション 5550  
BASIC-インタープリター

Printed in Japan N: SX18-2200-1

The IBM logo, consisting of the letters 'IBM' in a bold, sans-serif font, with each letter formed by eight horizontal bars of varying lengths. A registered trademark symbol (®) is located at the bottom right of the logo.

日本アイ・ビー・エム株式会社

東京都港区六本木3-2-12 〒106

TEL(03)586-1111 (代)