

# 第3章

## RX-78とコミュニケーション

この章では、「ダイレクトモード」「プログラムモード」となんだかやたら難しそうな言葉が出てくる。でも、おじけづいてはいけない。

大事なことがいっぱいあるが、無理していっどんに覚え込むことはない。何度も間違えながら、そのたびに読み直して確認していってほしい。

# 新発見! RX-78

## ダイレクトモードで、RX-78と熱い仲がうまれる

### 点を打つ

さあ、これから RX-78 とのコミュニケーションがはじまる。まず、画面をきれいにした状態からはじめよう。**SHIFT** を押しながら、**CLR HOME** を押す。そして、つぎのように打ってみよう。

SET 70, 150

これは画面に点を打つ命令だ。しかし、RX-78 は反応してこない。命令を画面に打ちこんだだけでは、RX-78 にはつたわらない。命令を読みこませる、つまり「入力」が必要なのである。

### 「入力」からはじまる RX-78との対話

入力する、とは、**RETURN** を押すこと。押してみれば、ほら、画面に点が出てくる。<sup>(注1)</sup>同じ方法でもうひとつ点を打ってみよう。今度はどこに点が出てくるだろうか。

SET 160, 130 **RETURN**

画面に点が出ると同時に、Ready という文字が出てくる。これは、RX-78 が準備OKという意味で、「さて、つぎは?」と君に聞いかけてているのだ。

### 円を描く

今度は、この2つの点を中心とした円を描いてみよう。

CIRCLE 70, 150, 30 **RETURN**

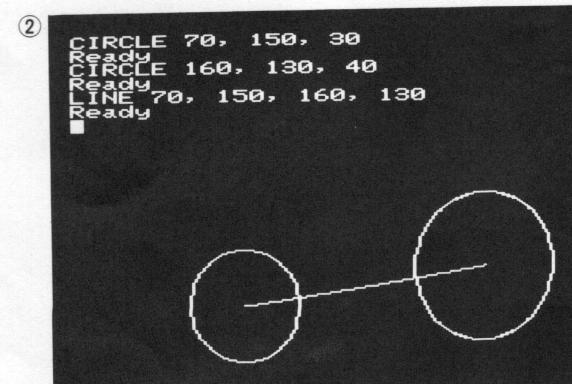
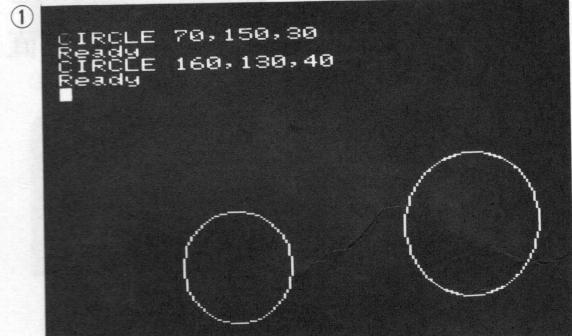
CIRCLE 160, 130, 40 **RETURN**

円が2つ描けただろう。(写真①)

### 線を引く

さらに、この2つの円の中心を直線で結んでみよう。

LINE 70, 150, 160, 130 **RETURN** (写真②)



<sup>(注1)</sup> **RETURN** を押す時に、入力させる行にカーソルがあるかどうか確かめましょう。

# さあいよいよさあざ對話はRX-78

## 「音」と「計算」

「点を打て」、「円を描け」、「線を引け」と命令するたびに、RX-78がその命令に応じてくれる。君とRX-78のこの熱い関係を、ダイレクトモードという。ダイレクトモードは、このほかいろいろなことができる。音を出すこともできるし、電卓以上の複雑な計算も可能。たとえば、つぎのように命令してみよう。

MUSIC "CEG" RETURN

「ド・ミ・ソ」と音が出る。

また「上底20cm、下底30cm、高さ15cmの台形の面積」は

PRINT (20+30) \* 15/2 RETURN

で、たちどころに解決してしまう。

このように RX-78 と対話をするために「BASIC」ということばを覚える必要がある。ここから RX-78 の世界は広がりをもちはじめる。ダイレクトモードは、「BASIC」への入口だ。さあいよいよ、RX-78 がおもしろくなってくる。



# RX-78が個性をもちはじめる

## ダイレクトモードをこえるプログラムモード

ダイレクトモードの RX-78だけでは物足りない、少しばかり不満を感じてきたらしめたもの。君はBASICの入口に近づいたことになる。

### プログラムモードとは

コンピュータと対話するには、君の考えがしっかり整理されて、伝えられなければならない。「何をするか」「どういう順番でするか」ということが、コンピュータにわかるようにつくられた命令の組みあわせ、これをプログラムという。

プログラムを組んで、コンピュータと対話する。これをプログラムモードとよぶ。これによって、コンピュータは2つ以上の命令をうけとめはじめる。

### ダイレクトモードの命令に順番をつけてみる

さてダイレクトモードに順番をつければどうなるだろう。その前に

NEW RETURN

と入力しなければならない。NEWとは、RX-78の頭の中をカラッポにする命令だ。

さて、頭の中がスッキリしたところで、次のプログラムを入力してみよう。

```
10 CIRCLE 70, 150, 30 RETURN
20 CIRCLE 160, 130, 40 RETURN
30 LINE 70, 150, 160, 130 RETURN (写真①)
```

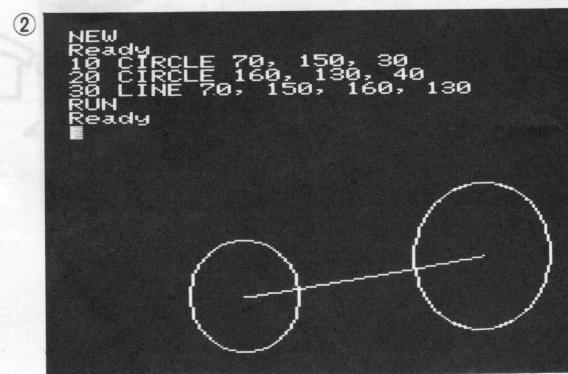
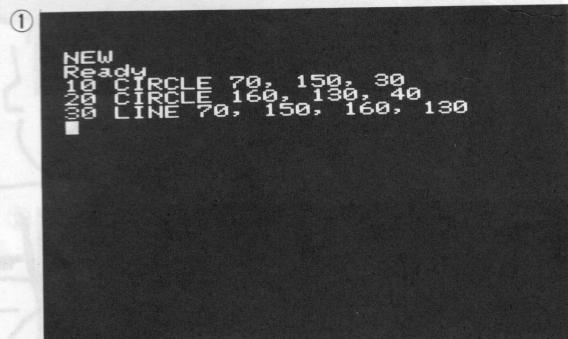
RX-78は反応してこない。RETURNを押して入力したはずなのに

### 走れ、を忘れている

RX-78は君の命令をまだ待っている。だから、走れ、という君の気持ちを伝えなければ動かない。その合図が

RUN RETURN

である。入れてみれば、RX-78が動きだした。(写真②)



# （フロン）で命令文を区切れば

## プログラムは行番号順に動きだす

RX-78が動く順番は、プログラムの頭につけられた番号順だ。これを行番号という。1~65,535までの整数(1,2,3,...,65535)が使える。命令は、行番号の小さい順からはじまってゆく。RX-78との約束のひとつだ。

## 図形と音がドッキングした。

たとえばこう入力してみよう。

```
15 MUSIC "C" [RETURN]
25 MUSIC "E" [RETURN]
35 MUSIC "G" [RETURN]
```

できたかな？

では、

RUN [RETURN]

RX-78が動きだした。

- ①まず円がひとつできて「ド」の音
- ②次にもうひとつ円ができる「ミ」の音
- ③最後に円の中心を線で結んで「ソ」の音

思わず納得してしまう。もう一度！(写真①)

RUN [RETURN]

さっきの絵と重なってくる。見にくいときは消してしまうのだ。

**SHIFT** を押しながら **CLR HOME** を押せば消えることは前にもいった。  
もう一度

RUN [RETURN]

## RX-78の記憶をたどる

いま目の前の RX-78 に

LIST [RETURN]

と入力してみてほしい。君が RX-78 に入力したものが、行番号順にでてきた。(写真②) 行番号をつけて入力したものは、何度も君の命令をくり返してくれる。あとから命令の追加もできる。そして、命令文を直したり、消したりすることもできる。

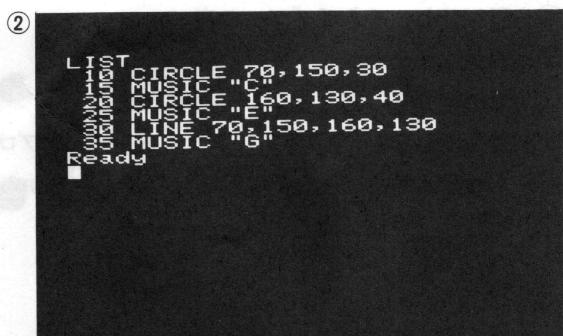
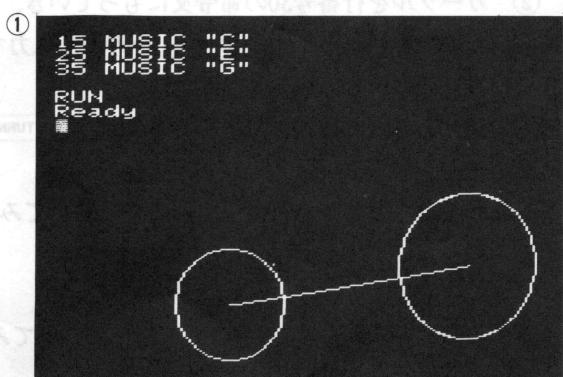
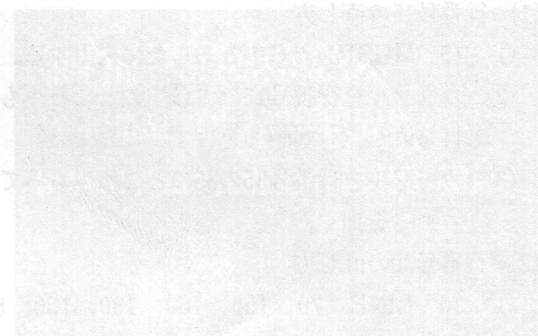
注1. 行番号には、1.5といった小数、3/5といった分数は使えません。

注2. ここでは **NEW** [RETURN] を入れないでください。

注3. **RESET** ボタンを押したり、電源スイッチを切ると、プログラムはすべて消えてしまします。

# 支那語でやること

```
10 CIRCLE 20, 120, 30
11 MUSIC "G"
30 CIRCLE 160, 130, 40
32 MUSIC "E"
38 MUSIC "C"
39 MUSIC "G"
40 LINE 70, 150, 160, 130, 40
41 CIRCLE 160, 130, 40
42 MUSIC "E"
43 MUSIC "C"
44 MUSIC "G"
45 CIRCLE 160, 130, 40
46 MUSIC "G"
47 CIRCLE 160, 130, 40
48 MUSIC "E"
49 MUSIC "C"
50 MUSIC "G"
51 CIRCLE 160, 130, 40
52 MUSIC "G"
53 MUSIC "E"
54 MUSIC "C"
55 MUSIC "G"
56 CIRCLE 160, 130, 40
57 MUSIC "G"
58 MUSIC "E"
59 MUSIC "C"
60 MUSIC "G"
61 CIRCLE 160, 130, 40
62 MUSIC "G"
63 MUSIC "E"
64 MUSIC "C"
65 MUSIC "G"
66 CIRCLE 160, 130, 40
67 MUSIC "G"
68 MUSIC "E"
69 MUSIC "C"
70 MUSIC "G"
71 CIRCLE 160, 130, 40
72 MUSIC "G"
73 MUSIC "E"
74 MUSIC "C"
75 MUSIC "G"
76 CIRCLE 160, 130, 40
77 MUSIC "G"
78 MUSIC "E"
79 MUSIC "C"
80 MUSIC "G"
81 CIRCLE 160, 130, 40
82 MUSIC "G"
83 MUSIC "E"
84 MUSIC "C"
85 MUSIC "G"
86 CIRCLE 160, 130, 40
87 MUSIC "G"
88 MUSIC "E"
89 MUSIC "C"
90 MUSIC "G"
91 CIRCLE 160, 130, 40
92 MUSIC "G"
93 MUSIC "E"
94 MUSIC "C"
95 MUSIC "G"
96 CIRCLE 160, 130, 40
97 MUSIC "G"
98 MUSIC "E"
99 MUSIC "C"
```



## プログラムを直す

前に入力したプログラムをつぎのようにしてみよう。

```
10 CIRCLE 70, 150, 30
15 MUSIC "G"
20 CIRCLE 160, 130, 40
25 MUSIC "E"
30 LINE 70, 150, 160, 130, 150, 80, 70, 150
```

行番号15、30の命令文が少し変わり、行番号35の命令文がなくなつた。さて、どう直すか。この通り全部入力しなおすという手もあるが、もっと簡単な方法がある。

### (1) 行番号15の直し方

- ① 15 MUSIC "G" とあらたに入力する。
- ② カーソルを行番号15の"C"のところへもってゆき、"G"と打ち直し、**RETURN** を押す。
- ③ カーソルを行番号35の35のところへもってゆき、15と直し  
(注1)  
**RETURN** を押す。

### (2) 行番号30の直し方

- ① 30 LINE 70, 150, 160, 130, 150, 80, 70, 150 とあらたに入力する。
- ② カーソルを行番号30の命令文にもっていき、150, 80, 70, 150と行の終わりに追加し、**RETURN** を押して入力する。

### (3) 行番号35の消し方

- ① 35とあらたに行番号だけを打ちこみ、**RETURN** を押す。

直し終わったら、ちゃんと直っているか確かめてみよう。  
(注2)

**LIST** **RETURN**

うまくいっているかな。(写真①)では、走らせてみよう。

**RUN** **RETURN**

どう?さっきと違って

- ① 円がひとつできて「ソ」の音
- ② 円がもうひとつできて「ミ」の音
- ③ 最後に2つの円と重なる三角形

が出てきたかな。(写真②)これで、RX-78がプログラムの通りに、しかも命令の順番に従って動いていることがわかつてもらえたと思う。

注1. 行番号35の命令文はそのまま消えないで残っています。

注2. 一番やりやすい方法で直してみましょう。

前に入力したプログラム

```
10 CIRCLE 70, 150, 30
15 MUSIC "C"
20 CIRCLE 160, 130, 40
25 MUSIC "E"
30 LINE 70, 150, 160, 130, 150, 80
35 MUSIC "G"
```

①

```
LIST
10 CIRCLE 70, 150, 30
15 MUSIC "G"
20 CIRCLE 160, 130, 40
25 MUSIC "E"
30 LINE 70, 150, 160, 130, 150, 80
,70, 150
Ready
```

②

```
LIST
10 CIRCLE 70, 150, 30
15 MUSIC "G"
20 CIRCLE 160, 130, 40
25 MUSIC "E"
30 LINE 70, 150, 160, 130, 150, 80
,70, 150
Ready
RUN
Ready
```

# : (コロン)で区切ってマルチステートメント

## : (コロン)で命令文を区切れば

命令文に行番号をつけてRUN [RETURN] とやれば、行番号の順に実行されることがわかったが、じつは、1つの行に2つ以上の命令文を書くこともできる。

たとえば、次の2つのプログラムは同じものになる。プログラム①で行番号10, 20, 30としていたのを、プログラム②では、行番号10の1行だけで3つの命令を行なわせている。

(プログラム①)

```
10 CIRCLE 70, 150, 30
20 CIRCLE 160, 130, 40
30 LINE 70, 150, 160, 130
```

(プログラム②)

```
10 CIRCLE 70, 150, 30 : CIRCLE 160, 130, 40
      : LINE 70, 150, 160, 130
```

つまり、命令文と命令文を:(コロン)で区切れば、1つの行番号でたくさんの命令文を実行することができる。<sup>(注)</sup>

ダイレクトモードでもこの方法は使え、次のようにもできる。

```
LINE 100, 60, 100, 180 : LINE 50, 120, 150, 120
:CIRCLE 100, 120, 50 [RETURN]
```

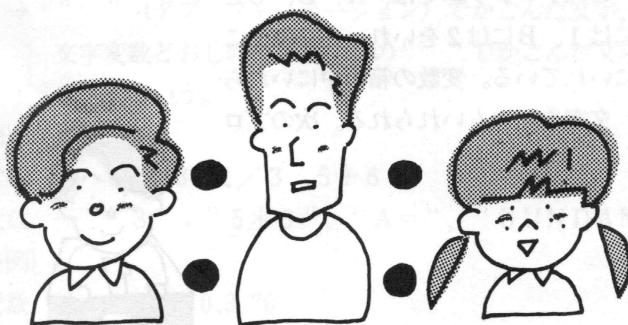
①

```
10 CIRCLE 70, 150, 30
20 CIRCLE 160, 130, 40
30 LINE 70, 150, 160, 130
10 CIRCLE 70, 150, 30 : CIRCLE 160, 130, 40
      : LINE 70, 150, 160, 130
LINE 100, 60, 100, 180 : LINE 50, 120, 150, 120
:CIRCLE 100, 120, 50
```

## マルチステートメントとは

このように、:(コロン)を使って1つの行にたくさんの命令文や計算式を書くことを、マルチステートメントという。マルチステートメントは、うまく使わないと、プログラムが見づらくなり、修正が非常にめんどうになってしまうことがある。

:(コロン)の使い方もプログラム作りの重要なテクニック。プログラム作りのうまい人が、どんなふうに:(コロン)を使っているか、そんな点も見るのがしてはいけない。



① 1つの行番号で書ける命令文は画面3行分までで、3行をこえる場合は、新たに行番号をつけなければなりません。

## 何かと便利な変数

ベーシックマスターBOOK(CD)

### 変数を味方にしよう

BASICでプログラムを作るには、便利で強力な味方がある。それは、「変数」とよばれるものだ。次のプログラムを入力してもらいたい。

```
NEW      RETURN
10 A=1    RETURN
20 B=2    RETURN
30 C=A+B  RETURN
40 PRINT A RETURN
50 PRINT B RETURN
60 PRINT C RETURN
```

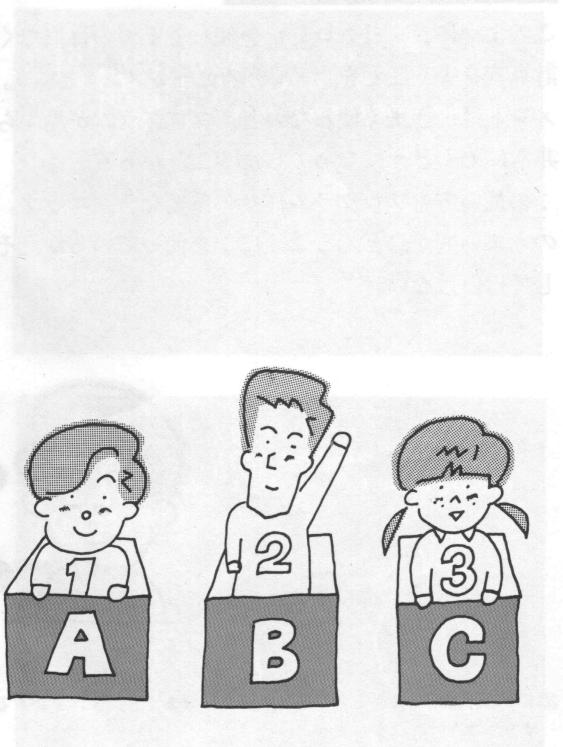
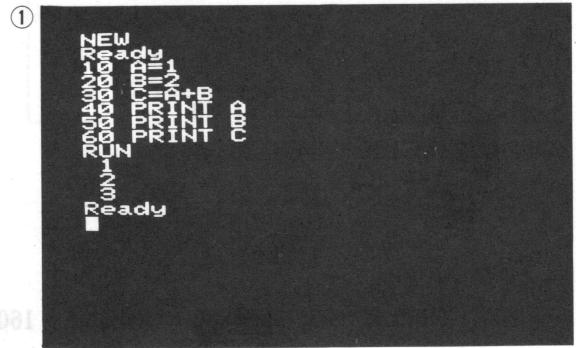
これは、「A、B、Cを画面にだしなさい」というプログラムなのだが、このプログラムを動かしても、A、B、Cとはでてこないで、1、2、3とでてくる。  
なぜこうなるのだろうか。このプログラムには次のような意味があるのだ。

NEW	→ 前のプログラムを消しなさい。
10 A=1	→ Aは1です。
20 B=2	→ Bは2です。
30 C=A+B	→ CはAとBをたしたものです。
40 PRINT A	→ Aのなかみを画面にだしなさい。
50 PRINT B	→ Bのなかみを画面にだしなさい。
60 PRINT C	→ Cのなかみを画面にだしなさい。

行番号30、40、50の命令文で、それぞれ、「A」、「B」、「C」ではなく、「Aのなかみ」、「Bのなかみ」、「Cのなかみ」を画面にだしなさい、といっているので、結果はそれぞれ、1、2、1+2の答3となるわけだ。

### 変数は箱だ

今まで見たA、B、Cは、それぞれ変数だ。変数というのは箱であると考えると、わかりやすい。このプログラムでは、A、B、Cという3つの箱を用意して、Aには1、Bには2を入れ、さらに、このふたつをたしたものを、Cにいれている。変数の箱の中にいれられるのは、数字だけではない。文字や記号もいれられる。次のプログラムをためしてみよう。



①プログラムを動かすというのは、RUN [RETURN]と入力することです。写真①

## 文字や記号は文字変数

```

NEW      RETURN
10 S $ = "*****"    RETURN
20 N $ = " GUNDAM ! "  RETURN
30 B $ = " BANDAI - "  RETURN
40 S S $ = S $ + S $  RETURN
50 BN $ = B $ + N $  RETURN
60 PRINT S S $  RETURN
70 PRINT BN $  RETURN
80 PRINT S S $  RETURN

```

さあおうと号記入式(英字半角)モードで莫じてokで実行して下さい。

食器棚(千草町)②

```

LIST
10 S $ = "*****"
20 N $ = "GUNDAM!"
30 B $ = "BANDAI-"
40 S S $ = S $ + S $
50 BN $ = B $ + N $
60 PRINT S S $
70 PRINT BN $
80 PRINT S S $
Ready
RUN
*****
BANDAI-GUNDAM!
*****
Ready

```

動かすとどうだろう、すなおに感動だ。(写真②)

前でてきた、数字をいれることができる変数A、B、Cなどを数値変数といい、S \$、N \$などの、文字や記号を入れることができるもの変数を文字変数という。

変数は魅力的で、たのもしい味方だから、プログラム作りに役立てもらいたい。同時に、変数を使うには、守らねばならないルールがある。

### 変数のルール

**[変数の名前]** ●英大文字(A~Z)ではじまる英数字のならび。ただし、文字変数は最後に\$が必要。

●3文字以上の場合、最初の2文字までを識別できる。たとえば、TATEとTAKASAは、はじめの2文字が同じTAなので同じ変数としてつかわれる。

●BASICの命令語と同じ名前、または命令語を含む名前は使えない。

#### (使える例)

数値変数 → A, T, C1, K10, TATE, NAME

文字変数 → F \$, P \$, D3 \$, YOKO \$

#### (使えない例)

数値変数 → 1A, 10S, E#, Gア, a, Hm, GO, IF, XRUN

文字変数 → 30 \$, ウ\$, ?L \$, NEWS\$, ABSET \$

**[変数のなかみ]** ●数値変数 → 数字および計算式

●文字変数 → " " (ダブルクォーテーション)でかこんだ文字、記号のならび、および文字変数どおりのたし算。この" "でかこんだ文字、記号のことを「ストリング」という。

#### (使える例)

数値変数 → 5, 0.3, 1/3, 5+6

文字変数 → "3", "5\*6", "A=", "GUNDAM", A \$ + B \$

#### (使えない例)

数値変数 → "5", 0.3 %

文字変数 → A, 2+3, "ワタシハ"GUNDAM"デス" (" "の中に、"をい  
れることはできない。)

## 計算記号の約束を覚える

RX-78 のかけ算やわり算の記号は、数学で使う計算記号とはちがっている。ここで、RX-78 で使う計算記号（算術演算子）を紹介しておこう。

### ●算術演算子

	計算記号	RX-78 の計算記号
たし算	+	+
ひき算	-	-
かけ算	×	*
わり算	÷	/
べき乗	$A^B$	$A \uparrow B$
左カッコ	(, [, {	(
右カッコ	), ], }	)
負号	-	-

### ●計算例

計算式	RX-78 での計算式
$(13+5) \times 7$	$(13+5)*7$
$21 \div 5 \times 10^2$	$21/5*10 \uparrow 2$
$\frac{1}{3} + \frac{2}{5}$	$1/3 + 2/5$
$\frac{2}{7} \times \frac{3}{4} \div \frac{2}{3}$	$(2/7)*(3/4)/(2/3)$
$-5 + 6 \times (-3)^2$	$-5 + 6*(-3) \uparrow 2$
$\{(13-5) \div 2 + 5\} \times 6$	$((13-5)/2+5)*6$
$(-6)^2 - 5^2$	$(-6) \uparrow 2 - 5 \uparrow 2$

RX-78 で使う計算記号のうちで、一番注意したいのは=（イコール）である。ふつう、 $A=B$ と書いた場合、A と B が等しいという意味だけど、RX-78 では「A に B をいれる」という意味になる。

### ●=を使った例

例	意味
$A = 5 + 3$	$5 + 3$ の答、8 を A にいれる
$X = X + 1$	$X + 1$ の答を X にいれる
$N \$ = " ガンダム "$	N \$ にガンダムといれる
$A \$ = " 5 + 3 "$	A \$ に $5 + 3$ といれる

RX-78 ではまた、大きな数字や小さな数字を下表のように10のべき乗を使って表わすことができる。

### ●大きな数字や小さな数字の表わし方

数 字	数学では	RX-78 では
300,000,000	$0.3 \times 10^9$	.3E+09
0.000036	$0.36 \times 10^{-4}$	.36E-04
-45,000,000,000	$-0.45 \times 10^{11}$	-.45E+11
-0.0000057	$-0.57 \times 10^{-5}$	-.57E-05

ただし、使える数字は、次の範囲である。

$\pm 0.27105055E-19 \sim \pm 0.92233720E+19$



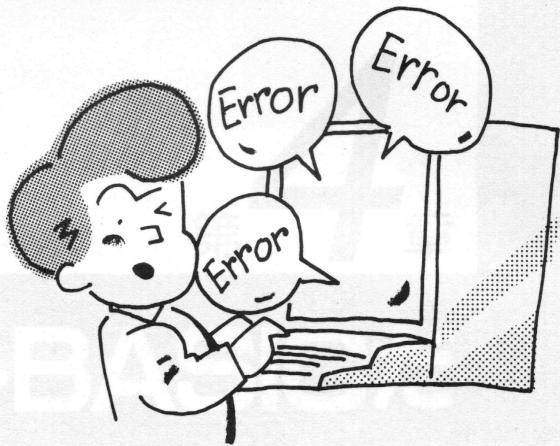
# エラーなんてこわくない

## エラーはつきもの

プログラムを作っているときに、命令文をまちがえたり、キーボードからの入力をまちがえたりすることがある。こうしたまちがいをエラーという。エラーがおこると RX-78 は、ピッという音といっしょに、画面に

\*Error 1 in 10

というようなメッセージをだしてくる。エラーを何回やっても、RX-78 はこわれない。しかし、エラーばかりだしてては、先に進まない。そんな時、もう一度画面のメッセージを見てほしい。



## エラーメッセージで解決

エラーメッセージは、ただ単に命令文がエラーであると言っているだけではない。どこでエラーになったのか、なぜエラーなのかということまで教えてくれている。

エラーメッセージのくわしい内容については、161ページの「ドクターレコード RX-78」にまとめて書いてある。さあ、エラーメッセージを見てプログラムを直しながら、プログラム作りの腕に磨きをかけよう。

# RX-78が記憶する

## カセットレコーダと組む

プログラムモードで命令文をいれてやれば、それを RX-78 が記憶<sup>(注1)</sup>して、何度も実行することができる。しかし、電源を切ってしまうと、RX-78 はすっかり忘れてしまう。やっとでき上ったプログラムや、作りかけのプログラムが、一瞬のうちに消えてしまうのはいただけない。これを解決するには RX-78 とカセットレコーダをつなげばよい。

これで君のプログラムのひとつひとつが、テープに記憶されてゆき、プログラムのカセットライブラリーを作ることだってできるのだ。

注1. プログラムモードとは行番号つきの命令文のことです。

注2. カセットレコーダのつなぎ方は「取扱説明書」を、テープに記憶させる RX-78 への命令は、116ページの「カセットは頭脳カプセル」を見てください。