

はじめての PB

～初級 PB-120 入門～

●滝本飛沫

PB-120 って何ですか？

あなたは PB-100 を知っていますか？ 最近、始めて手にしたポケコンが PC-E500 だ、という人も多いようですから、知らない人もいるでしょうね。PB-100 は CASIO 社が昔に発売していたポケコンです。そう、PB-100 は昔のポケコンなのです。しかし、この名前は現在も見ることができます。それというのも、PB-100 の後継機が、今なお頑張っているからです（もちろん PB-100 自体も頑張っています）。

PB-100 シリーズは、PB-110 の時代に、一度 BASIC のコマンドに変更を加えましたが、それ以後も PB-100 との上位互換を保ち続けました。PB-120 はその中の 1 つです。

PB-120 は、外見上 PB-110 と良く似ています。中身も、BASIC には、ほとんど違いが見られません。PB-120 の一番の特徴は、7,648 ステップという記憶容量にあります。それまで、同型の PB シリーズは、標準で 544 ステップ、RAM を増設して 1,568 ステップだったのですから、みなさんの発売時の驚きは大変なものだったでしょう。

現在、CASIO 社にとって PB-120 は、「BASIC プログラム入門機」として位置づけされているようです。PB-120 は、他

の大型ポケコンに比べ、機能がシンプルで使いやすく、低価格なのが特色です。

● スイッチ ON から始めよう ●

PB-120 に付属の＜入門書＞は、本体の使い方を大変分かりやすく説明してあります。今回は、初心者を対象としているので、＜入門書＞と重複する部分もありますが、PB-120 を使うために必要な基礎事項を書いていきたいと思います。

まずは、電源をいれましょう。本体左側で、ディスプレイの下にあるスイッチを ON 側(右側)にスライドさせると、電源が入ります (図 1)。



図 1 電源投入時の画面

この時、画面上側に“RUN”と点灯しているのは、今の状態が「RUN モード」であることを示しています。「RUN モード」では、マニュアル計算や、プログラムの実行ができます。

● 書き込みモードに切り換えよう ●

前の状態から **MODE** **1** と押すと、「WRTモード」になります (図2)。



図2 WRTモードの画面

画面上側の“WRT”は、今の状態が「WRTモード」であることを示しています。「WRTモード」では、プログラムを記憶させることができます。

「P 0123456789」は、各プログラム・エリアの使用状況を示していて、プログラムが入力されていないときは数字が、入力されているときは“_”が表示されます。

画面上側の「7648」という小さな数字は、残りステップ数です。PB-120では、プログラムの記憶容量を「ステップ」で数え、プログラムを記憶させる度に、残りステップ数が減っていきます。

● プログラム入力前に心の準備 ●

ここからプログラムを書き込むことができるのですが、その前に心の準備が必要です。

もし、WRTモードにしたときに図2のようにならなかった場合(プログラム・エリアに数字ではなく“_”が表示された場合)、そのエリアには何かプログラムが入っていることになります。これからプログラムを入力するのに、この様な状態だったときには、次のいずれかを選択しなければなりません。

- a. 残りステップ数に余裕があり、プログラム・エリアの空きがある場合、空

きエリアにプログラムを書き込む。

- b. 残りステップ数に余裕がないか、プログラム・エリアに空きがない場合で、今入っているプログラムを消去しても良いと思う場合、現在記憶しているプログラムを消去し、新しくプログラムを入力する。
- c. 残りステップ数に余裕がないか、プログラム・エリアに空きがない場合で、今入っているプログラムを消去したくない場合、新しくプログラムを打ち込むことはあきらめる。

実際には、もっと様々な場合があり、色々な対処法がとれるのですが、今回はbの場合で話を進めます。

N **E** **W** **A** **L** **L** **EXE**

と入力してください。これで画面が図2のようになるはずです。

● プログラムを打ち込もう ●

まずは、画面に“A”を表示するプログラムをプログラム・エリア0に入力してみましょう。

1 **0** **SHIFT** **Z** **SHIFT** **W**

A **SHIFT** **W** **EXE**

(または **1** **0** **P** **R** **I** **N**

T **SHIFT** **W** **A** **SHIFT** **W**

EXE)

上の通り打ち込むと、下のようなりストが入力されたことになります。

10 PRINT "A"

以後、リストはこのように表記します。

EXE はリスト上で書かれることはありませんが、画面上に書かれたプログラムを PB-120 に記憶させるために押すキーです。行の最後には、忘れずにこのキーを押してください。

なお、プログラムを打ち込む際に、キーの押し間違いがあったときは、次の要領で訂正してください。

- a. **EXE** を押す前に、間違いに気付いた場合、そのままカーソル・キーやデリート・キー、インサート・キーを用いて、間違えた所を書き直す。
- b. **EXE** を押した後に、間違いに気付いた場合、LIST **EXE** と入力した後、カーソル・キーやデリート・キー、インサート・キーを用いて、間違えた所を書き直す（訂正した後、忘れずに**EXE** を押すこと）。

※カーソル・キー、デリート・キー、インサート・キーの使用例

例 1 A のつもりで Z を押してしまった

```
10 PRINT "Z"
```

カーソルキー（**←** **→**）を押してカーソル“**_**”を間違えたところ（「Z」のところに）合わせ、正しい文字（「A」）を上書きする。

例 2 A を 2 回押してしまった

```
10 PRINT "AA"
```

カーソル・キーでカーソルを間違えたところ（余分な「A」のところに）合わせ、

デリート・キー（**DEL**）を押し、余分な文字を消す。

例 3 A を書き忘れた

```
10 PRINT ""
```

カーソル・キーでカーソルを挿入したいところ（後ろの「**"**」のところに）に合わ

せ、インサートキー（**SHIFT** **DEL**）で 1 文字分スペースを開け、挿入したい文字（「A」）を書き込む。

● プログラムの実行 ●

さて、プログラムを入力し終わったら、次は実行です。**MODE** **0** と押すと、図 1 の画面になります。ここから **SHIFT** **0** と押すと、プログラム・エリア 0 に入っているプログラムが実行されます。



図 3 プログラムの実行画面

“A”というのは、PRINT という命令によって画面に表示された文字です。上の小さな「STOP」という文字は PRINT 文によって、実行が一時停止していることを示します。

これは、「画面上に“A”を表示させる」プログラムですから、これで終わってしまいます。しかし、これでプログラムを打ち込んで、実行させることができるようになったわけです。

● BASIC と変数 ●

先ほどのリストをもう一度見てみましょう。

```
10 PRINT "A"
```

先頭の数字は「行番号」といって、プログラムの実行順序を示しています。次の、「PRINT」は命令で、PB-120 に対して仕事を指示するものです。BASIC プログラムは、このように、行番号と命令によって構成されます。

次に、プログラムを作る際の重要な要素「変数」について説明します。

変数というのは、数値や文字を記憶させておく場所のことで、PB-120 の場合、通常 26 か所あります。その場所には、それぞれ A から Z までの名前がつけられています。

変数は、その中に入れるものが数値である場合「数値変数」、文字である場合「文字変数」と呼ばれ、文字変数は「A\$」のように、変数名に「\$」を付けて数値変数と区別します（「\$」という専用文字変数も存在します）。なお、「A」と「A\$」のように、同じアルファベットで示される数値変数と文字変数は、データを記憶する場所が同じ位置になるので、どちらか一方しか使えません。

ところで、変数にはもう 1 つ、

「A(1)」のように変数名に添字を付けて変数につながりを持たせる「配列変数」があります。PB-120 の場合、A から Z までを、一連の配列として使えます。これは「A」と「B」は、そのままでは別々の変数なのですが、「A」を「A(0)」「B」を「A(1)」としてつながりのある配列変数にすることができるのです。配列変数は、主に、大量なデータを記憶したり、処理したりするときに使います。

● 初歩の BASIC ●

ここからは、プログラムを動かしながら、BASIC の命令がどういう働きをしているか、プログラムがどのように動くのかについて、見ていきましょう。

リスト 1

```
10 X=0
20 PRINT CSR X;"A"
30 X=X+1
40 IF X<12 THEN 20
50 END
```

リスト 1 を、WRT モードで書き込み、RUN モードで実行してください。

最初、左端に“A”が表示されます。ここで EXE を押すと、今度は左から 2 番目に“A”が表示されます。その後 EXE を何度か押すと、“A”が左から右へと動いていき、右端に来るとプログラムが終了します。

まずここで使われている命令について 1 つずつ見ていきましょう。

※代入文(LET)

10 行では X と 0 が、30 行では X と X+1 が、それぞれ「=（イコール）」で結ばれています。この「=」は数学の「=」と違い、あくまで右辺の式の値を左辺の変数に入れる（代入する）役目をします。このような文を代入文といいます。

BASIC の命令には LET という、代入のための命令がありますが、省略することができるので通常 LET を付けずに代入文のみ書かれます。

※IF~THEN

IF に続く式が成り立つとき、THEN 以下の文を実行し（文ではなく数字の場合は、その数字で示される行へ実行を移します）、成り立たなかったとき、THEN 以降は無視します。

40 行の場合、式が「 $X < 12$ 」なので、 X が 12 以上か未満かで判定します。 X が 12 未満のときは、式が成り立つので、THEN 以降を実行します。この場合、20 という数字が書かれているので、20 行に飛びます。また、 X が 12 以上のときは、式が成り立たず、THEN 以降は無視して、次の行を実行します。

※END

プログラムを終了させます。

※PRINT

PRINT に続く数値、文字、変数の内容などを画面に表示します。

※CSR

PRINT 文中に使われ、表示位置を指定します。

20 行の場合、変数 X の値で示される位置に文字を表示します。

● プログラムの動きを追う ●

個々の命令の説明が終わったので、次はこのプログラムが、どのように動いているかを見ていきましょう。RUN モードで SHIFT 0 または RUN EXE と入力すると、プログラムが先頭から始まります。リスト 1 場合、先頭は 10 行です。

まず、10 行を見ると、代入文の「 $X=0$ 」があります。これにより数値変数 X の値は 0 になります。

次の 20 行は PRINT 文です。 X の値で

示される位置に“A”を表示させます。今の X の値は 0 です。つまり画面上の 0 の位置（左端）に“A”を表示することになります。

そのとき、プログラムは一時停止します。なぜなら PB-120 の PRINT には「表示後一時停止し、EXE が押されたなら、一度画面をクリアし、次へ進む」という働きがあるためです。ここでは EXE が押されたものとして、次へ進みます。

30 行はまた代入文です。

変ですね。「 $X=X+1$ 」では左辺と右辺とが同じ（イコール）ではありません。しかし、BASIC の場合、これで良いのです。この文は、「 X の内容を取り出してそれに +1 し、元の X に入れなさい」という意味で、右辺の X は古い X 、左辺の X は新しい X なのです。ここで X には、「前の $X+1$ 」つまり「 $0+1$ 」で答えの「1」が入ります。

40 行は IF 文です。今の X の値は、先ほどの代入文で、1 になっています。IF 文の条件式は「 $X < 12$ 」なので、今の X の値は条件を満たしています。従って、THEN の後ろを実行します。ここでは、「20」という数字が書かれているので、20 行へ実行を移します。

また、20 行です。今度は X の値が 1 なので、“A”は、画面上の 1 の位置（左から 2 番目）に表示されます。ここでまた一時停止をして、EXE が押されると、30 行にすすみます。

30 行では、 X の値が「 $X+1$ 」つまり、「 $1+1$ 」で 2 になります。

40 行で判定をすると、今度も X が 12 よりも小さく、THEN の後ろを実行し、20 行へ実行を移します。

このようにして、 X が 1 つずつ増えていき、“A”の表示位置も右の方へ動いていきます。そして、 X が 11 になったとき、20

行で“A”を画面上の11の位置（右端）に表示し、30行でXが12になり、40行の判定で、「 $X < 12$ 」に当てはまらなくなります。すると、こんどはTHENの後ろが無視され、50行が実行されます。

50行はENDです。つまり、ここでこのプログラムは終了します。

これが、リスト1の動きです。

● プログラムを作ろう ●

今度は右から左に“A”が動くプログラムを作ってみましょう。

リスト2

```
10 X=11
20 PRINT CSR X;"A"
30 X=X-1
40 IF X≥0 THEN 20
50 END
```

何だかリスト1に似ていますね。それというのも。今度のプログラムは“A”の移動方向が右から左に変わっただけなので、始めのXの値や、Xに加える値、判定の式が変わるだけのなのです。ということかということ「キャラクタが動く」というのは、「表示位置が変化する」であり、それはつまり「変数の内容が変わる」ということなのです。「キャラクタの動きが変わる」というのは、「表示位置の変化の仕方が変わる」つまり「変数の内容の変化の仕方が変わる」ということになります。

リスト2もリスト1のときと同じ様に、プログラムの実際の動きを追ってみてください。

今度は“A”が右から左へ動きます。RUNしてみると、画面上でもその通り動くはずで

プログラムというのは、この様に「変数」が重要な働きをします。プログラムを作りたいと思うなら。人の作ったプログラムをPB-120に打ち込み、動かし、リストの流れ（プログラムの動き）を追ってみましょう。始めは短い物がいいでしょう。ゲーム・プログラムの場合、少しひねくれた処理をしているかもしれませんが。マニュアルを見ても分からないところは「何だこんなおかしい物作りやがって」などと笑い飛ばして。打ち込んだゲームで遊んでしまえば良いのです。それを何度も繰り返すうちに、自然にBASICの命令や使い方を覚えて行くものです。

PB-120は良いポケコンです。使い方さえ覚えてしまえば、プログラムぐらい造作もなく作れるようになる。そんな機種です。普段、上着のポケットに入れて持ち歩き、必要に応じて、必要な処理をその場で作りすぐに使う。そんなポケコンにするのはあなた次第です。

工学者PJ 1991年4月号 掲載
2023/05/14 校正 滝本飛沫