

IchigoJam での文字列の扱い

●ポインタ

IchigoJam のプログラムの中では、「"」(ダブルクォーテーション)で囲んで文字列を使うことができます。例えば、

```
PRINT "ABC"
```

```
PLAY "CDE"
```

また、変数に代入することもできます。

```
A="ABC"
```

この時、変数 A にはどんな値が入るのでしょうか？

以下の 2 行のプログラムを入力して、実行してみましょう。

```
10 A="ABC"  
20 ?A  
RUN  
3078  
OK
```

変数 A には何だか変な値が入っていますね。

20 行を改造して、16 進数で表示してみましょう。

```
10 A="ABC"  
20 ?HEX$(A)  
RUN  
C06  
OK
```

この値は、「ABC」という文字列が入っているプログラム領域(メモリ)のアドレスです。

このように、アドレスの値が入っている変数を「ポインタ」あるいは「ポインタ変数」と言います。

●IchigoJam のメモリマップ

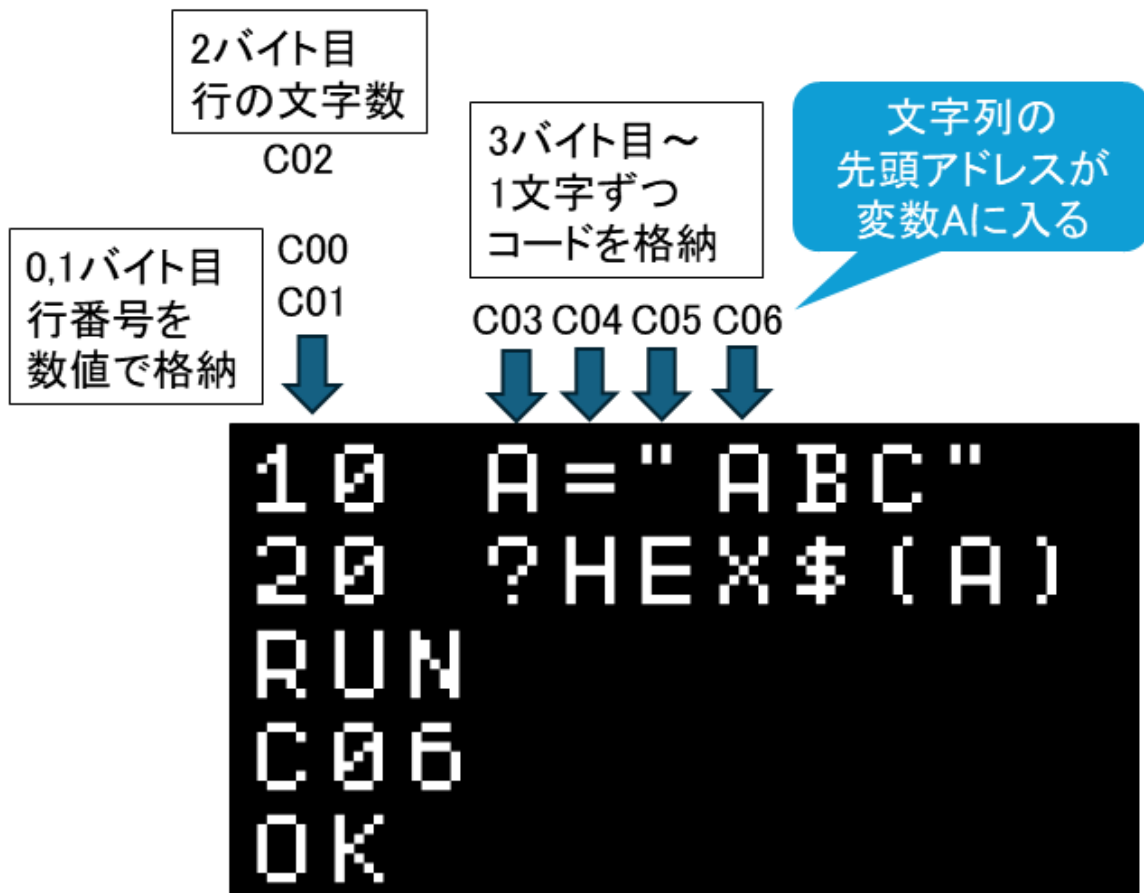
IchigoJam のメモリは、右図のような
割り当てになっています。

さきほどプログラムで表示した A の値は、

```
10 A="ABC"
20 ?HEX$(A)
RUN
C06
OK
```

プログラム領域にある、
文字列の先頭アドレスを表しています。

アドレス	内容
#000	キャラクターROM 文字コード0~223 (書き込み不可)
#6FF	
#700	PCG領域 文字コード224~255 (書き換え可能)
#7FF	
#800	配列変数[0]~[101] 各2バイト×102個
#8CB	
#8CC	通常変数A~Z 各2バイト×26個
#8FF	
#900	画面VRAM 32文字×24行=768バイト
#BFF	
#C00	プログラム領域 1Kバイト
#FFF	



●ポインタを利用したプログラム

プログラムの 20 行を、STR\$()を使って書き換えてみましょう。

```
10 A="ABC"
20 ?STR$(A)
RUN
ABC
OK
```

STR\$()は、IchigoJam のリファレンスでは「PRINT 内で、文字列を返す」と説明されていますが、要は「()内の値をポインタとして、そのアドレスのメモリから格納されている文字列を表示する」コマンドです。

STR\$()は表示する文字数も指定できるので、それを使ってプログラムを改造してみましょう。

```
10 A="ABC"
15 FOR N=0 TO 2
20 ?STR$(A+N,1)
30 NEXT
RUN
A
B
C
OK
```

このように、文字列の一部を取り出して表示することができます。

新しいプログラムを作って、PLAY 命令を試してみましょう。

```
10 P="C;D;E;F;G;A;B;" 音階の文字列のアドレスを P へ
20 K=INKEY():IF !K CONT 何かキーが押されるまで待つ
30 K=K-49 1~7の数字キーを0~6の数値に変換
40 PLAY P+K*2 K*2番目の文字列を演奏
50 GOTO 20 20行へ戻る
RUN
```

キーボードの1~7の数字キーを押すと、ド~シの音が鳴ります。

つまり通常は「PLAY "CDE"」などと文字列で音階を指定しますが、実はポインタ変数でも指定ができて、「ポインタが示すアドレスにある文字列を演奏する」という動きをします。