

LESSON

6




ネコをつかまえる




キー入力で操作しよう

表示する位置を変えよう

次のプログラムを入力しましょう。行の最後に  を押すのを忘れないようにしてください。  を押して実行すると、 が画面中央に表示されます。

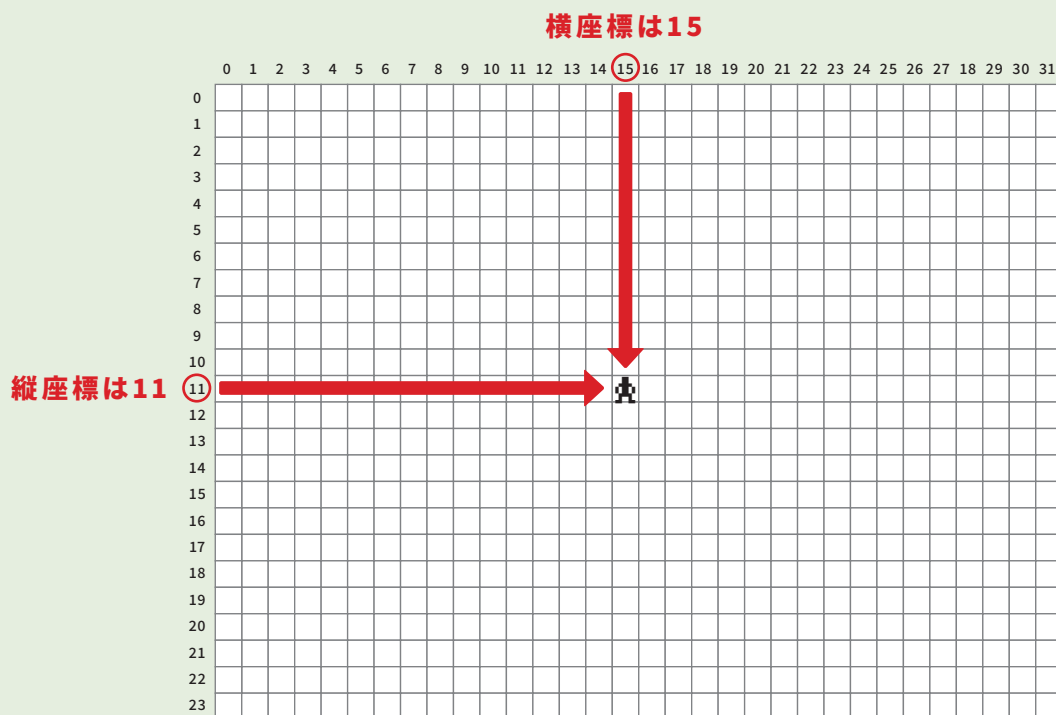
```
10 CLS  
70 LOCATE 15,11:PRINT"
```

 は、エンターキーを押す合図です。

ロケイト

LOCATE命令は、文字や絵文字を表示する位置を指定するための命令です。

LOCATE 横座標, 縦座標



考えよう!



♣をひとつ右の位置に表示するには、プログラムをどのように変更したらよいでしょうか？

答え


70行を次のように変更しましょう。 **F4 LIST** キーを押してプログラムを表示します。カーソルキーを使ってカーソルを「」まで移動し、5を消して6を入力し、**Enter** を押します。

```
70 LOCATE 16,11:PRINT"♣"↵
```

F5 RUN

を押して確認してみましょう。

今度は、プログラムを次のように変更してください。

それぞれの行を変えたら、行ごとに必ず  を押しましょう。

```
10 CLS:X=15
70 LOCATE X,11:PRINT"★"
```

F5
RUN

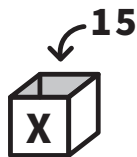
を押して確認してみましょう。最初と同じ位置に★が表示されます。

考えよう!



70行は変更せずに、★をひとつ右の位置に表示するには、10行をどのように変更したらよいでしょうか？

答え



へんすう
Xは変数です。変数には、数を入れて覚えさせておくことができます。

70行でLOCATE命令の横座標にXを指定していますね。Xには15が入っているので、横座標には15が指定されていることになります。

LOCATE  , 11

そこで、10行のX=15をX=16に変えれば、70行のプログラムはそのまま、ひとつ右に表示されますね。



LOCATE  , 11

このように、LOCATE命令で変数を使うと、変数の中身を変えることで、位置を調整できます。変数のアルファベットは何を使ってもよいのですが、横座標にはXを、縦座標にはYを用いることが多いです。

キー入力で位置を変えよう

次のように、20行、40行、100行のプログラムを追加しましょう。

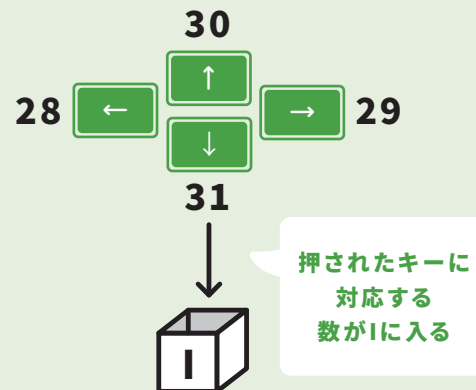
```
20 I=INKEY()↵
40 IF I=29 X=X+1↵
100 GOTO20↵
```

F5 RUN を押して実行すると、カーソルキーの **→** を押すたびに、**▲**が右に増えていきます。

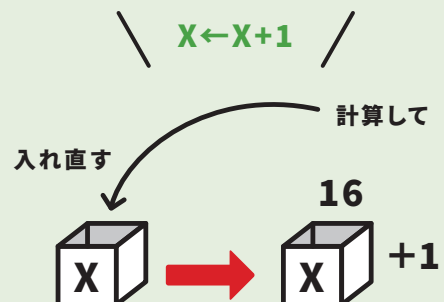


うまく動かないときは、**ESC** を押して実行を止めて、**F4 LIST** を押してプログラムを表示し、20行、40行、100行のプログラムに間違いがないかよく確認しましょう。


インキー
INKEY()は、キーボードのどのキーが押されているか調べる命令です。
右矢印キーが押されている場合、変数 **I** には29が入ります。何も押されていなければ0が入ります。



40行のIF(もしも)によって、**I**に29が入っているときだけ、**X=X+1**が実行されます。
X=X+1は、変な式ですね。この**=**は、左向きの矢印のように考えてください。




最初、Xには16が入っていますから、 $X=X+1$ すると、Xの中身は17に変わります。
Xが17になると、70行のLOCATEによって、横座標17の位置に▲が表示されます。

このように、を押すたびにXは17、18、19…と変わっていき、▲がどんどん右に表示されていきます。

残像を消そう

次のプログラムを追加してみましょう。


```
90 LOCATE X,11:PRINT" "
```

“ ”の間は、スペースキー（キーボードの一番下の大きなキー）を一度押しましょう。


70行とほぼ同じプログラムですが、▲の代わりにスペースを表示します。スペースは**何も表示しないという文字**です。▲を表示したあとに同じ位置にスペースを表示すると、▲は消されます。

この場合、コンピューターはすごく速いので、▲を表示してすぐ消してしまい、▲がほとんど見えませんね。消す前にほんの少し待つようにしてみましょう。

```
80 WAIT 3
```

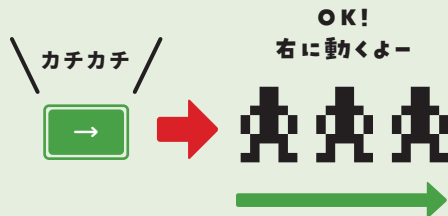
で確認してください。残像を消しながら位置を変えて表示するので、▲が増えずに、移動するようになりましたね。

左にも動かそう

を押してみてください。▲が左に動きますか？動きませんね。なぜでしょう。
左に動かすプログラムを作っていないからです。

人間なら、気を利かせて左に動いてくれるかもしれませんが。でもコンピューターは、人間が指示したこと以外は何もしてくれません。

右の矢印キーを
カチカチしてみると...



左の矢印キーを
カチカチしてみると...



左にも動くように、次のプログラムを追加しましょう。□の部分には、何が入るでしょうか？

```
30 IF I=28 X=X□1
```

縦にも動かそう

左右だけでなく、上下にも動かせるようにしてみましょう。

次のプログラムを追加して変数Yに縦座標を入れておき、上下のキー入力があった場合、1増えたり減ったりするようにします。

```
50 IF I=30 Y=Y-1
60 IF I=31 Y=Y+1
```

最初は画面の中央に表示されるように、10行でYに11を入れておきましょう。

```
10 CLS:X=15:Y=11
```

70行を次のように変えます。これまでLOCATE命令の縦座標に11を指定していたので、上下には動きませんでした。11の代わりにYを使うことで、変数Yの中身が変われば表示される位置が変わり、上下に動くようになります。

```
70 LOCATE X,Y:PRINT"▲"↵
```

F5
RUN

を押して実行してみましょう。

おや？何か変ですね。上下に動かすと、また残像が残るようになってしまいました。

／ 考えよう！ ／

残像を消すプログラムをなおすのを忘れました。どこをどうなおしたらよいか考えて、プログラムを編集してみてください！

ここまでの
プログラム



縦横で動くかな？

```
10 CLS:X=15:Y=11↵
20 I=INKEY()↵
30 IF I=28 X=X-1↵
40 IF I=29 X=X+1↵
50 IF I=30 Y=Y-1↵
60 IF I=31 Y=Y+1↵
70 LOCATE X,Y:PRINT"▲"↵
80 WAIT3↵
90 LOCATE X,Y:PRINT" "↵
100 GOTO20↵
```

画面の一番下までいくとスクロールしてしまうので、いかにないようにしましょう。

時間があまったら1:ゲームにしよう

▲を上下左右に動かすプログラムができたので、これを利用したゲームを考えてみます。

画面のどこかに現れるネコをつかまえるゲームなら、わりと簡単にできそうです。

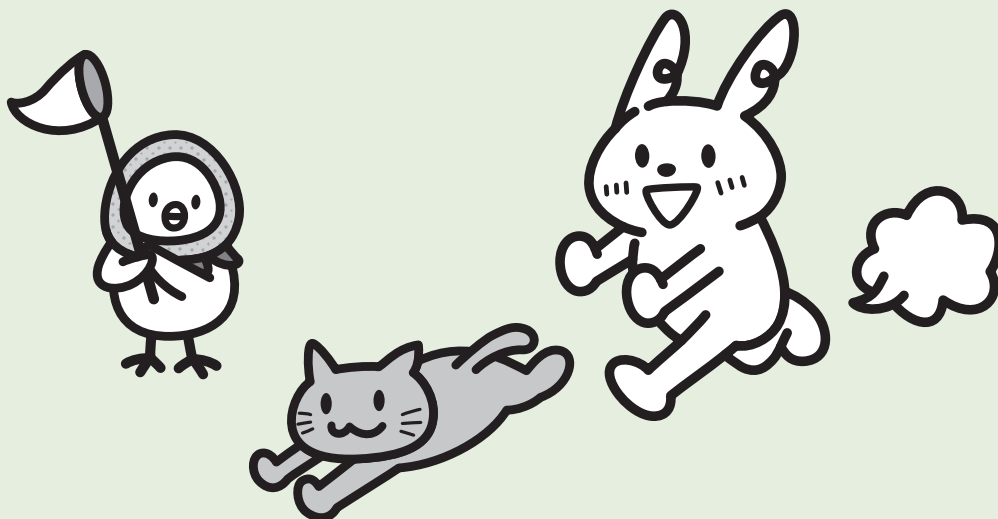
次のプログラムを入力すると、画面のランダムな位置にネコが現れます。▲を動かして捕まえると、また別のところにネコが現れます。

```
15 CLT:S=0:GOSUB110☐  
65 IF SCR(X,Y)=ASC("☺") GSB110☐  
120 LOCATE RND(32),RND(22)☐  
130 PRINT"☺":BEEP25,30:RETURN☐
```

☐はAltキーを押しながらCを押します。

さらに、次のプログラムを入力しましょう。何匹捕まえたか数字が表示され、10秒たったら終了するようになります。時間内に何匹捕まえらるか、挑戦してみましょう！

```
95 IF TICK()>600 END☐  
110 LOCATE0,22:PRINT S:S=S+1☐
```



時間があまったら2:難しい書き方に見よう

30行から60行までのプログラムを、少し短く書く方法があります。慣れたらこちらのほうが簡単かもしれません。

```
30 X=X-(I=28)+(I=29)↵
40 Y=Y-(I=30)+(I=31)↵
```

(I=28)は、Iが28のとき1に、それ以外のときは0になります。

(I=29)は、Iが29のとき1に、それ以外のときは0になります。

そこでもIが28だったら、30行は $X = X - 1 + 0$ となりXが1減ります。同じようにIが29だったら、 $X = X - 0 + 1$ となりXが1増えます。どちらでもないときはXは変わりません。Yも同じようにして、短く書くことができます。試してみましょう。

これらの行を入力したら、50行と60行を消しましょう。次のように、行番号と


Enter

だけを押すと、その行が消えます。

```
50↵
60↵
```

時間があまったら3:BTN命令を使ってみよう

ボタン

INKEY命令の代わりに、BTN命令を使っても、を動かすことができます。

BTN命令のほうが、よりスムーズに動きます。試してみましょう。

```
30 X=X-BTN(28)+BTN(29)↵
40 Y=Y-BTN(30)+BTN(31)↵
```