プチコンで横スクロールシューティングゲームを作る（１）

プチコンは、NINTENDO 3DS で BASIC（ベーシック）プログラムを作るソフトです。

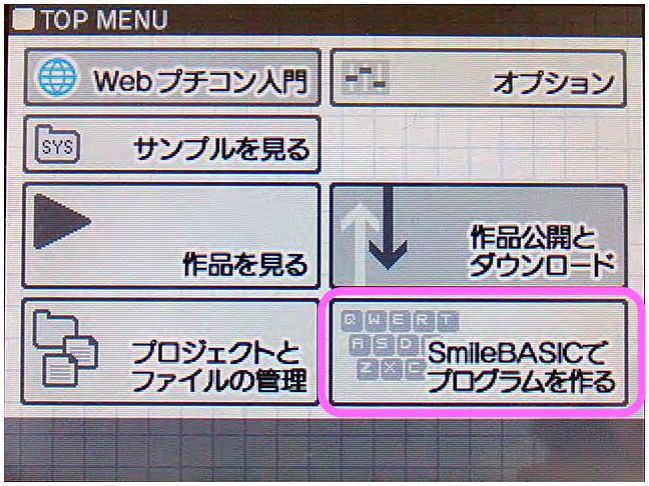
●今回の目標

プチコンで横スクロールのシューティングゲームのプログラムを作ります。
自機が画面左側に表示されて、右向きへ進んでいきます。
敵が右側から出てくるので、ビームを打って倒していきます。

●プチコンの基本操作

3DS のメニューから、「プチコン 3 号」を動かします。

ソフトが動くと、トップメニュー画面が表示されます。
下画面の「SmileBASIC でプログラムを作る」を選びます。



プログラムを実行する画面（DIRECTモード）が表示されます。

上画面が実行画面、下画面がキーボードです。
下画面でキーをタッチすると、上画面にその文字が入力されます。

試しにいろいろな文字を打ってみましょう。

キーボードで入力する文字の種類を変えるには、以下のようにします。

- 普通に打つ…大文字アルファベット「ＡＢＣ」
- SHIFT キーを押してから打つ…小文字アルファベット「ａｂｃ」
- 左下の文字きりかえキーで「♥」を押す…グラフィック文字が打てる。SHIFT キーでさらにきりかえ。
- 左下の文字きりかえキーで「ア」を押す…カタカナ・ひらがな文字が打てる。SHIFT キーでさらにきりかえ。

文字を消す時は、バックスペース（BS）キー  またはデリート（DEL）キー  を使います。

- バックスペースキー…カーソルの前の文字が消える
  Ａ｜ → ｜
- デリートキー…カーソルの後ろの文字が消える
  ｜Ａ → ｜

●DIRECT モードでプログラムを動かす

まずは DIRECT モード画面で、かんたんなプログラムを打って、動かしてみましょう。

```
PRINT "コンニチハ"
```

「PRINT」の後ろは、空白（スペース）を1文字空けます。
「コンニチハ」の両側は「"」（ダブルクォーテーション）で囲みます。
上のプログラムを打って、Enter キーを押すと、次の行に「コンニチハ」と表示されます。

```
PRINT "コンニチハ"
コンニチハ
```

「**PRINT**」（プリント）命令は、画面に文字を表示する命令です。
文字をいろいろ変えて、表示してみましょう。

PRINT 命令では、計算もできます。
「PRINT」に続いて（1文字空白を入れて）計算式を書いて Enter キーを押すと、次の行に計算の答えが表示されます。

```
PRINT 1+1
2
```

いろいろな計算をしてみましょう。計算記号は以下のように書きます。

- 足し算…「**+**」
- 引き算…「**-**」
- かけ算…「*****」
- わり算…「**/**」

画面に文字がいっぱいになったら、「**ACLS**」（エーシーエルエス）命令を使うと、画面がクリアされます。

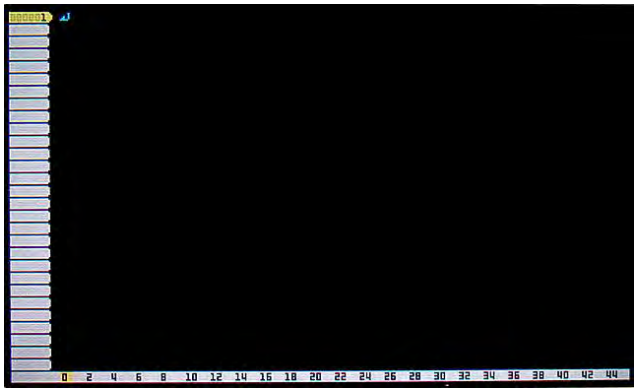
```
ACLS
```

●EDIT モードで長いプログラムを作る

DIRECT モードでプログラムを書くと、1 行しか書けません。
長いプログラムを作るには、以下の手順で行います。

① EDIT モードでプログラムを書く
② DIRECT モードでプログラムを実行する

キーボードの「EDIT」ボタンをタッチします。
上画面が EDIT モードの画面になります。
画面左側に「1」「2」…と行番号が表示されます。



少し長いプログラムを書いてみましょう。

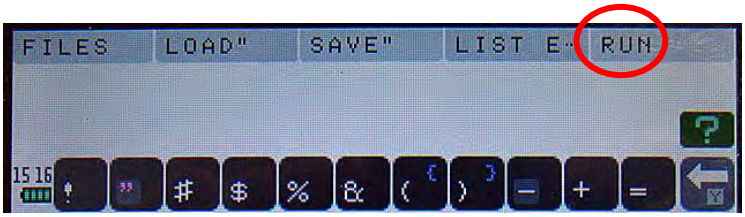
```
1 PRINT "コンニチハ"
2 PRINT "コンバンハ"
```

1 行目で「コンニチハ」を表示、2 行目で「コンバンハ」を表示するプログラムです。
まず、この 2 行を編集モード画面で打ちます。
次に、キーボード画面左下にある「実行」ボタンをタッチして、実行モードにします。
実行モード画面で「**RUN**」(ラン)と打って Enter キーを押すと、プログラムが実行されて、2 行の文字が表示されます。

```
RUN
コンニチハ
コンバンハ
OK
```

プログラムは、基本的に上から下へ順番に実行されます。
何行かプログラムを打って、実行してみましょう。

※プログラムを実行する命令「RUN」は、キーボード画面の一番上にあるファンクションキーの 5 番に登録されています。これをタップすると一度に入力できます。



●自機を表示する

プログラムの作り方の基本がわかった所で、いよいよゲームのプログラムを作っていきます。
まず、自機を画面に表示してみましょう。

★スプライトを表示する

プチコンには、「スプライト」というキャラクターを表示する機能があります。
スプライトを使って、自機を表示してみましょう。
編集モードの画面で、以下のプログラムを打ちます。

```
1 ACLS
2 SPSET 0,1260
3 SPSHOW 0
```

画面をクリア
管理番号 0 のスプライトを 1260 番に設定
管理番号 0 のスプライトを表示

実行すると、画面左上に自機が表示されます。

これだけだと、その前に表示していた文字などと重なってしまいます。
表示する位置を変えてみましょう。

```
1 ACLS
2 SPSET 0,1260
3 SPOFS 0,100,100
4 SPSHOW 0
```

管理番号 0 のスプライトの座標を(100,100)に設定

途中に新しい行を入れるには、行を入れたい場所にカーソルを移動して、キーボード右下の「L.INS」(行挿入)キーを押します。

プログラムを実行してみましょう。
画面の真ん中やや左に、自機が表示されます。

それぞれの命令の文法を説明します。
SPSET(エスピーセット)命令は、スプライトの種類を設定する命令です。
文法は、以下のようになっています。

```
SPSET        0          ,1260
          管理番号     定義番号
```

管理番号	スプライトをプログラム内で呼び出す時の番号。0～511 の範囲で自由に決められます。
定義番号	設定したいスプライトのキャラクター番号を指定します。(0～4095)

スプライトのキャラクター番号を調べる時は、以下のようにします。

- ✧ キーボードの「SMILE」ボタンを押す 

- ✧ SMILE ツール画面の下にある「SPDEF」ボタンを押す 

スプライトのキャラクターと番号が一覧で表示されます。

▲▼ボタンを押してスクロールすると、いろいろなキャラクターが表示されます。
3DS の「X」ボタンを押すと、編集画面にもどります。

SPSET 命令の定義番号(キャラクター番号)を変えると、表示されるスプライトが変わります。
いろいろ変えて試してみましょう。

スプライトの位置を変えるには、「SPOFS」(エスピーオフセット)命令を使います。

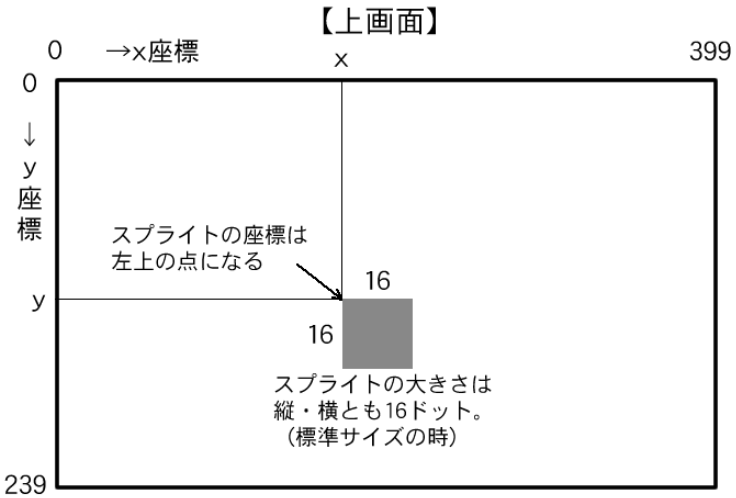
```
1 ACLS
2 SPSET 0,1260
3 SPOFS 0,100,100
4 SPSHOW 0
```

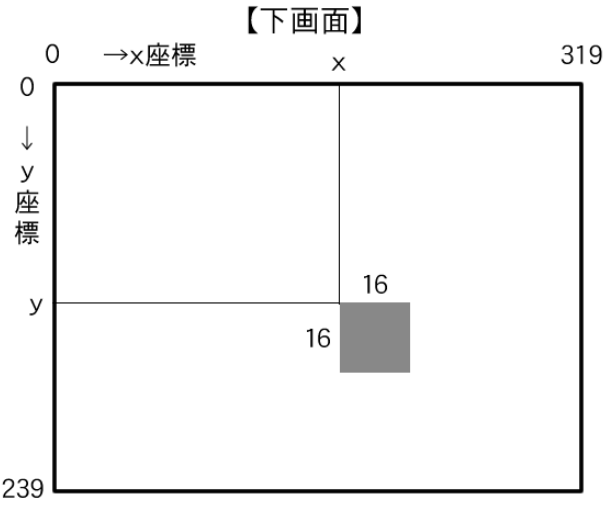
SPOFS 命令の文法は以下のようになっています。

SPOFS　0　,100　,100　,0
　　管理番号　x 座標　y 座標　z 座標

管理番号	スプライトの管理番号。0～511。
x 座標	スプライトのx座標。0～399。(画面の外でも受け付ける)
y 座標	スプライトの y 座標。0～239。(画面の外でも受け付ける)
z 座標 (省略可能)	スプライトの z 座標(前後)。1024(奥)～-256(手前)

プチコンの画面の座標は、右のようになっています。
画面のサイズは以下のようです。
【上画面】
　横:400 ドット　座標は 0～399
　縦:240 ドット　座標は 0～239
【下画面】
　横:320 ドット　座標は 0～319
　縦:240 ドット　座標は 0～239

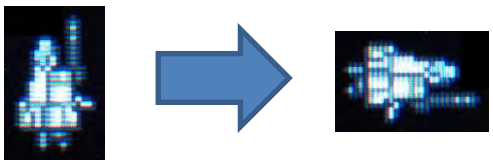




座標の数字を変えて、キャラクターをいろいろな位置に表示してみましょう。

「SPSHOW」(エスピーショー)命令は、指定した番号のスプライトを画面に表示する命令です。
「SPHIDE」(エスピーハイド)命令を使うと、そのスプライトを隠すことができます。

このままだと、自機が上向きになっています。
ゲームでは右へ進んでいきたいので、
自機を右向きに回転させましょう。



```
1 ACLS
2 SPSET  0,1260
3 SPOFS  0,100,100
4 SPHOME  0,16,16
5 SPROT  0,90
6 SPSHOW  0
```

スプライトの原点を中央に設定

スプライトを 90 度回転

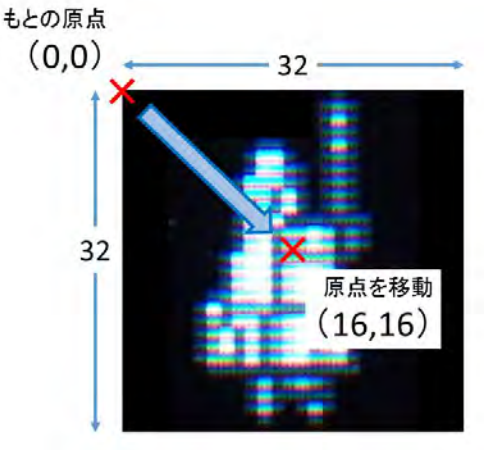
プログラムを実行してみましょう。
自機が右向きで表示されます。

SPHOME（エスピーホーム）命令は、スプライトの原点を設定する命令です。
文法は、以下のようになっています。

```
SPHOME      0          ,16          ,16
          管理番号    原点 x 座標    原点 y 座標
```

管理番号	原点を変更したいスプライトの管理番号（0～511）。
原点 x 座標	スプライトの原点の x 座標。
原点 y 座標	スプライトの原点の y 座標。

スプライトの原点はもともと左上ですが、あとでスプライトを回転させた時、そのままだとプログラムが難しくなるので、スプライトの中央へ原点を移動させます。
今回の自機のスプライトは 2 倍サイズ（32×32 ドット）なので、原点を中央にすると(16,16)になります。

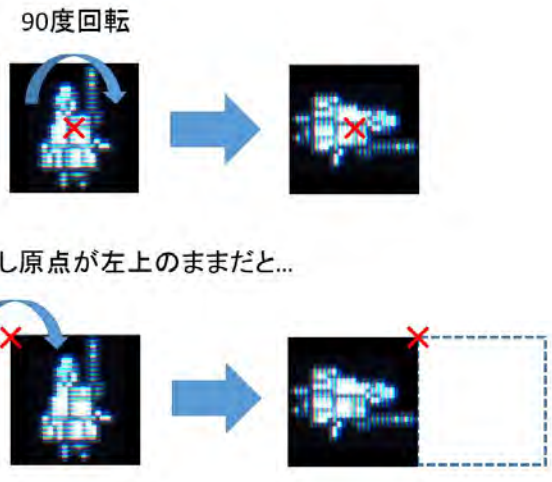


次に、SPROT（エスピーローテイト）命令で、スプライトを右に回転させます。

```
SPROT      0          ,90
        管理番号      回転角度
```

管理番号	回転させたいスプライトの管理番号（0～511）。
回転角度	原点を中心に回転させる角度。時計回りに 0～359。

これで自機のスプライトが 90 度回転して、右向きになります。



もし SPHOME 命令を使わず、原点がスプライト左上のままだと、90 度回転した後は、原点がスプライトの右上になってしまいます。
こうなると、後でスプライトの位置などがわかりづらくなるので、今回は SPHOME で原点を中心に移動しています。

●コメントを入れる

あとでプログラムを見る時に、どんなプログラムなのかわかりやすくするために、プログラムの中にコメントを入れることができます。
最初に、プログラムのタイトル「SHOOT1」を入れてみましょう。

```
1 '*** SHOOT1 ***
2 ACLS
3 SPSET 0,1260
4 SPOFS 0,100,100
5 SPHOME 0,16,16
6 SPROT 0,90
7 SPSHOW 0
```

プログラムのタイトル

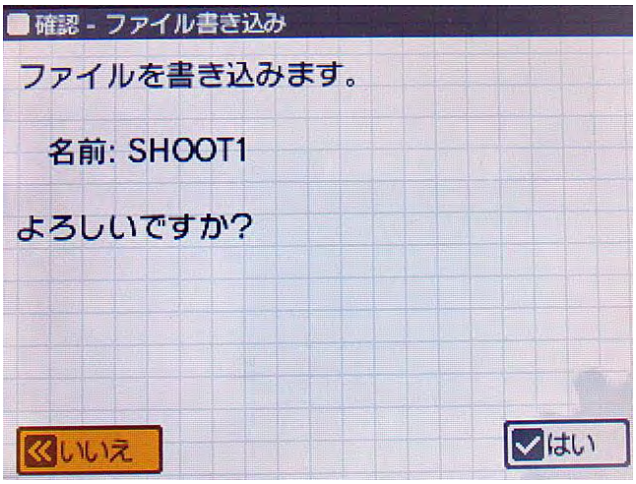
コメントを入れるには、先頭に「’ 」（アポストロフィ）をつけます。
「’ 」の後は、何を書いてもプログラムの実行には影響しません。
タイトル以外にも、ところどころでプログラムの説明を入れるといいでしょう。

●プログラムを保存する

このままだと、プチコンを終了させるとせっかく打ったプログラムが消えてしまいます。
保存をしておきましょう。
保存するには、DIRECT モードの画面で、「SAVE」(セーブ)命令を使います。

```
SAVE　”SHOOT1”
　　ファイル名
```

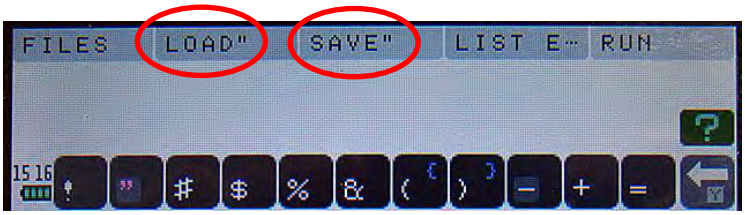
入力すると、下画面に確認画面が
表示されます。
「はい」をタッチすると、保存されます。



保存したプログラムは、「LOAD」(ロード)命令で呼び出せます。

```
LOAD　”SHOOT1”
　　ファイル名
```

LOAD 命令、SAVE 命令は、キーボードの上にあるファンクションキーの 2 番と 3 番に登録されているので、ワンタッチで入力できます。



●スコアと自機の残り数を表示

ゲーム画面らしく、スコア（点数）と自機の残り数を表示してみましょう。

これらはゲーム中にどんどん変化する数字なので、それを記憶するのに**変数**（へんすう）を使います。
変数は、算数で使う「□」（四角）と同じで、いろいろな数字を入れる箱のようなものです。
（中学の数学なら「x」や「a」などの文字と同じです）

今回はスコアを変数「SC」、自機の残り数を変数「PL」で記憶することにします。
それぞれ変数の値を設定して、画面に表示します。

```
 1 '*** SHOOT2 ***
 2
 3 '--- ガメン ---
 4 ACLS
 5 SC=0
 6 LOCATE 0,0
 7 PRINT "SCORE ";SC
 8 PL=3
 9 LOCATE 30,0
10 PRINT "LEFT ";PL
11
12 '--- ジキ ---
13 SPSET 0,1260
14 SPOFS 0,100,100
15 SPHOME 0,16,16
16 SPROT 0,90
17 SPSHOW 0
```

プログラムのタイトルを「SHOOT2」に

わかりやすくするために
空行とコメントを追加

スコアの変数 SC を 0 にする

カーソルの位置を(0,0)に

スコアを表示

自機の残り数 PL を 3 にする

カーソルの位置を(30,0)に

自機の残り数を表示

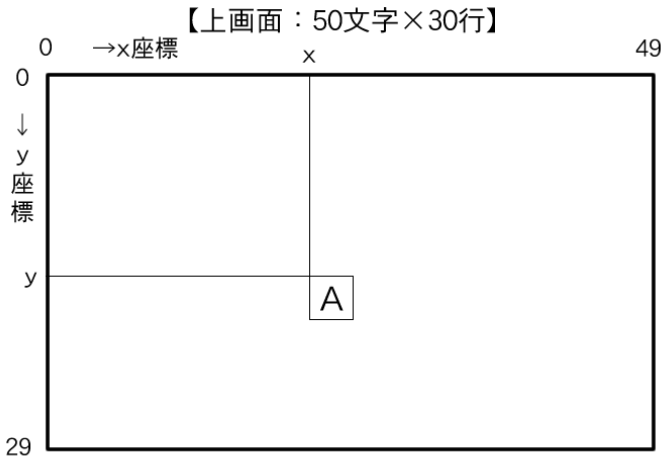
わかりやすくするために
空行とコメントを追加

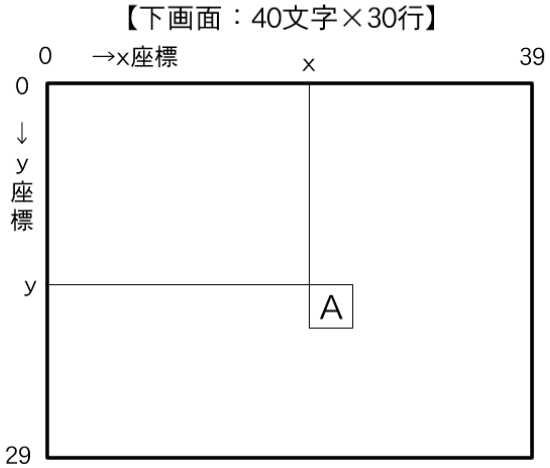
「SC=0」は、「変数 SC の値を 0 にする」という意味です。「SC と 0 が等しい」という意味ではないので、注意してください。

★LOCATE(ローケイト)命令

```
LOCATE      0          ,0
          x 座標      y 座標
```

文字を表示するカーソルの位置を設定します。
文字を表示する画面(コンソール画面)のレイアウトは、右の図のようになっています。





LOCATE 命令に続いて PRINT 命令を使うと、画面のいろいろな位置に文字を表示できます。
今回は「LOCATE 0,0」として、上画面の左上に文字を表示しています。

PRINT 命令で、文字に続けて変数の値を表示するには、「;」(セミコロン)で続けて変数名を書きます。今回は文字と変数の値の間を空けるために、スペースを 1 文字入れています。

★プログラムを「SHOOT2」の名前で保存しましょう。

```
SAVE " SHOOT2"
```

●自機をキーで動かす

自機を十字キーで上下左右に動かすプログラムを作りましょう。

★座標を変数で指定する

自機の横座標（x座標）と縦座標（y座標）を、変数「PX」「PY」で指定します。

```
 1 '*** SHOOT3 ***
 2
 3 '--- ガメン ---
 4 ACLS
 5 SC=0
 6 LOCATE 0,0
 7 PRINT "SCORE ";SC
 8 PL=3
 9 LOCATE 30,0
10 PRINT "LEFT ";PL
11
12 '--- ジキ ---
13 PX=100
14 PY=100
15 SPSET 0,1260
16 SPOFS 0,PX,PY
17 SPHOME 0,16,16
18 SPROT 0,90
19 SPSHOW 0
```

プログラムのタイトルを「SHOOT3」に

自機の座標(PX,PY)を設定

スプライトの座標を(PX,PY)に設定

この時点では、座標指定を変数に変えただけなので、プログラムを実行しても見た目は変わりません。

★キー入力を読み取る

自機を十字キーで動かすために、キーやボタンの入力を読み取る BUTTON（ボタン）関数を使います。
まずは BUTTON 関数の動きをテストしてみます。

```
12 '--- ジキ ---
  （中略）
18 SPROT 0,90
19 SPSHOW 0
20
21 '--- ジキヲ ウゴカス ---
22 @MOVELOOP
23 LOCATE 0,20
24 PRINT BUTTON(0);"  "
25 GOTO @MOVELOOP
```

コメントを入れる
もどってくるためのラベル
BUTTON 関数の値を画面に表示
@MOVELOOP へジャンプ

くり返しをするために、GOTO（ゴートゥー）命令を使います。

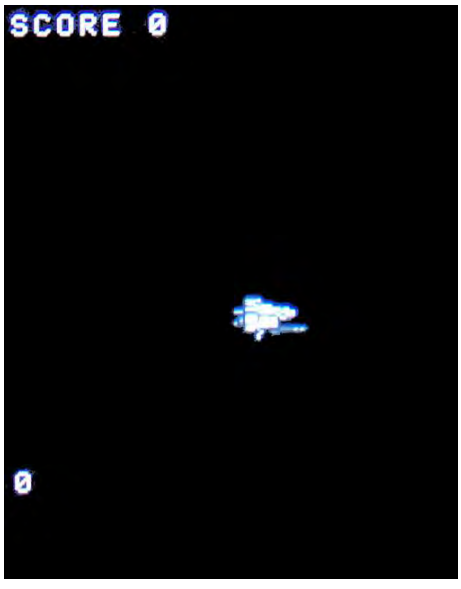
```
GOTO    @MOVELOOP
```

ジャンプ先ラベル

ジャンプする先は「**ラベル**」で指定します。
ラベルは、「@」（アットマーク）に続けてアルファベットや数字で指定します。

PRINT 命令では、BUTTON 関数の値をそのまま画面に表示しています。

プログラムを実行してみましょう。
画面の下の方に「0」と表示されます。
十字キーやボタンを操作してみましょう。数字がいろいろ変わるのがわかります。



BUTTON 関数の値は、押されたキーやボタンによって、以下の値になります。

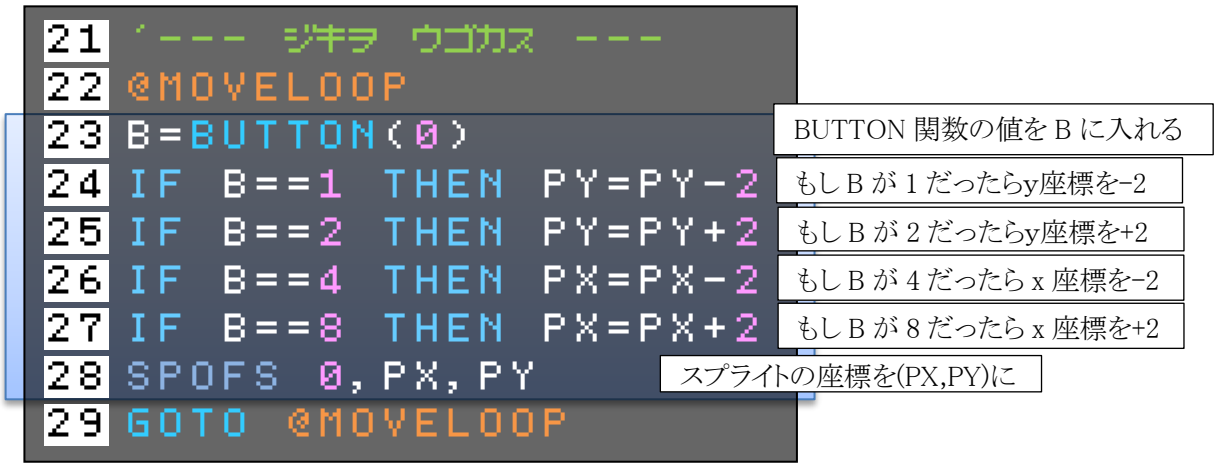
押されたボタン	値
何も押さない	0
十字キーの上	1
十字キーの下	2
十字キーの左	4
十字キーの右	8
A ボタン	16
B ボタン	32
X ボタン	64
Y ボタン	128
L ボタン	256
R ボタン	512

なお、2つのボタンを同時に押すと、それぞれの値を足し算した値になります。
例えば「十字キーの上」を押しながら「A ボタン」を押すと、1＋16＝17 になります。
十字キーやボタンをいろいろ押して、表の値になることを確認してください。

そのままだとプログラムがいつまでも終わらないので、キーボードの「STOP」キーを押して止めてください。

★十字キーで自機を動かす

BUTTON 関数の値を表示するプログラムを消して、自機を動かすプログラムを入力します。



まず、BUTTON 関数で十字キーを読み取ります。
何度もチェックしないといけないので、変数 B に値を入れます。

```
B=BUTTON(0)
```

BUTTON は関数なので、「=」で変数に値を入れる事ができます。

次に、「十字キーの上」が押されているかどうかを判断します。
判断するには「IF」(イフ)～「THEN」(ゼン)～「ELSE」(エルス)命令を使います。

```
IF B==1 THEN PY=PY-2
```

IF 命令の文法は以下のとおりです。

IF　B==1　THEN　PY=PY-2　ELSE　～

B==1	PY=PY-2	～
条件式	条件が成り立つ時に実行する命令	条件が成り立たない時に実行する命令

条件式	判断する条件の式を書く。式の例は以下のとおり。 「A==0」…A が 0 に等しい 「A<0」…A が 0 より小さい(0 はふくまれない) 「A>0」…A が 0 より大きい(0 はふくまれない) 「A<=0」…A が 0 以下(0 もふくまれる) 「A>=0」…A が 0 以上(0 もふくまれる) 「A!=0」…A が 0 に等しくない
THEN ～	条件が成り立った時に実行する命令を書く。
ELSE ～	条件が成り立たない時に実行する命令を書く。 ELSE 以下は省略可能。

あらためてプログラムに戻りますが、

```
IF B==1 THEN PY=PY-2
```

このIF命令は、「もし十字キーの上が押されていたら、y 座標を 2 減らす」になります。
プチコンの画面では、y 座標を 2 減らすと、上へ 2 ドット分移動することになります。
※プログラムで「X=X-1」は、「X と X-1 が等しい」ではなく、「X から 1 を引いた値を X に入れる」、つまり「X を 1 減らす」という意味になります。
同じように、十字キーの下・左・右が押された時のIF命令を書きます。

```
IF B==2 THEN PY=PY+2
IF B==4 THEN PX=PX-2
IF B==8 THEN PX=PX+2
```

(似たような行なので、コピー・ペーストするといいでしょう)
これで座標(PX,PY)が変えられたので、スプライトをその座標に移動します。

```
SPOFS 0,PX,PY
```

★行のコピー・ペースト

今回のように、同じようなプログラムの行を打つ時は、前に打ってある行をコピーすると楽です。
行をコピーするには、キーボード右下の「COPY」(コピー)・「PASTE」(ペースト)キーを使います。

① 十字キーでカーソルを移動して、コピーしたい行にカーソルが点滅した状態にする。

```
IF B==1 THEN PY=PY-2
```

② 「COPY」キーをタッチする。
③ 十字キーでカーソルを移動して、新しい行を入れたい場所へ持っていく。
④ 「PASTE」キーをタッチすると、同じ行が貼りつきます。

```
IF B==1 THEN PY=PY-2
```

⑤ 行の内容を書き換えます。

```
IF B==2 THEN PY=PY-2
```

このようにコピー・ペーストを使うと、最小限の書き換えですみます。

1 行だけではなく、複数の行をコピーしたい時は、「SEL」(選択)キーを押します。
選択モードになり、十字キーでカーソルを上下させると、複数の行が選択できます。
必要な部分を選択できたら、「COPY」キーでコピー、「PASTE」で貼りつきます。

さて、プログラムを実行してみましょう。
十字キーを押すと…自機が消えてしまいます。
プチコンのプログラムはものすごく速いので、十字キーを押すと、自機が画面の外へ飛んでいってしまうのです。
これではゲームにならないので、くり返しの中で時間待ち(ウェイト)を入れます。

```
'--- ジキヲ ウゴカス ---
@MOVELOOP
B=BUTTON(0)
IF B==1 THEN PY=PY-2
IF B==2 THEN PY=PY+2
IF B==4 THEN PX=PX-2
IF B==8 THEN PX=PX+2
SPOFS 0,PX,PY
VSYNC 1
GOTO @MOVELOOP
```

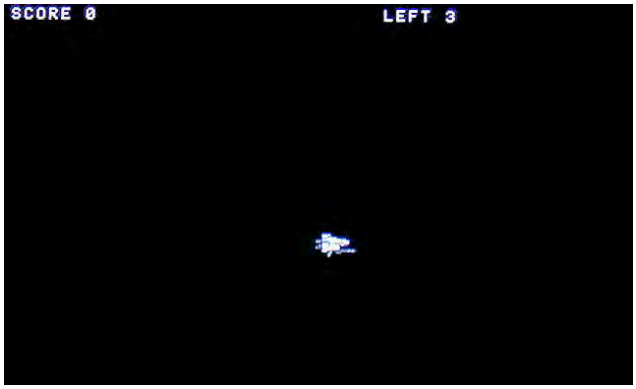
待ち時間を入れる

VSYNC(ブイシンク)命令は、時間待ちをする命令です。

VSYNC　　1
　　待ち時間

待ち時間	60 分の 1 秒単位。60 で 1 秒。

プログラムを実行してみましょう。
今度は自機が目に見える速度で動きます。

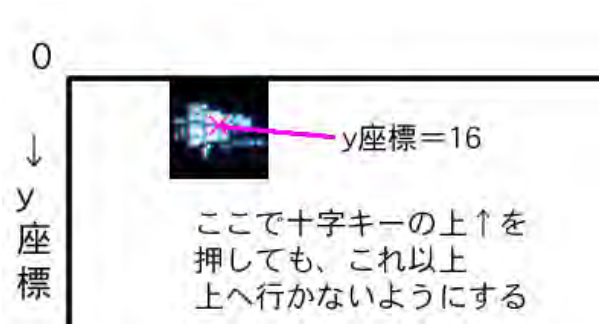


ただし、十字キーをずっと押していると、自機が画面の外へ出て行ってしまいます。
これではゲームにならないので、外へ出ないように条件を設定します。

```
'--- ジキヲ ウゴカス ---
@MOVELOOP
B=BUTTON(0)
IF B==1 AND PY>16 THEN PY=PY-2
IF B==2 AND PY<224 THEN PY=PY+2
IF B==4 AND PX>16 THEN PX=PX-2
IF B==8 AND PX<384 THEN PX=PX+2
SPOFS 0,PX,PY
VSYNC 1
GOTO @MOVELOOP
```

画面からはみ出さないように条件を追加

考え方としては、例えば自機のスプライトが画面の一番上にいれば、y座標が16（あるいはそれ以下）になるので、その時は上へ行かないようにします。



IF 命令の条件式に、「AND」（アンド）でつないで、2つの条件を入れます。

```
IF B==1 AND PY>16 THEN PY=PY-2
```

この条件式は、「B が 1 と等しい、かつ、PY が16より大きい」という意味になります。
両方の条件が成り立った時だけ、「THEN」以下の命令が実行されます。
（※「～かつ～」は、中学の数学の「集合」で習います。数式では「A∩B」などと書きます）

同じように、画面の下・左・右はじでもはみ出さないように、AND 条件式を入れます。

```
IF B==2 AND PY<224 THEN PY=PY+2
IF B==4 AND PX>16 THEN PX=PX-2
IF B==8 AND PX<384 THEN PX=PX+2
```

これで自機の動きは完成です。

※2つの条件をつなぐ条件式は、もう一つ「OR」(オア)があります。

```
IF B==1 OR PY>16 THEN PY=PY-2
```

この条件式は、「B が 1 と等しい、または、PX が 16 より大きい」という意味になります。
2つの条件のどちらかが成り立つと、「THEN」以下の命令が実行されます。
(※「～または～」も、中学の数学の「集合」で習います。数式では「A∪B」などと書きます)

★プログラムを「SHOOT3」の名前で保存しましょう。

```
SAVE " SHOOT3"
```