IchigoJam（いちごじゃむ）でプログラミング

センサースイッチ対応版

ミッションカード Mission POSSIBLE! CARD

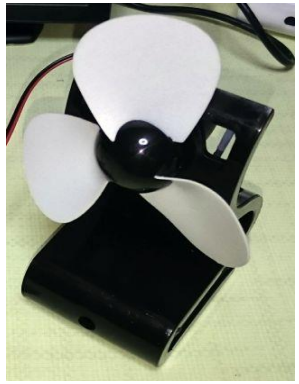
君だけの小っちゃなコンピュータ。

100円（平成元年） せんぷう機 を便利にまわそう！

100円ショップのせんぷう機をプログラムで、多機能に変身させます。

モノづくりとプログラミングのはじめの一歩！

おもしろい・たのしいからはじめよう！！



・本資料はNPO法人NEXTDAYの協力のもとNPO法人小樽青少年の科学の芽を育てる会が作成しました。
・資料の作成にあたり以下の資料を参照しました。
＞親子でベーシック入門 IchigoJamではじめてのプログラミング （出版社: ジャムハウス）
＞IchigoJamでプログラミング （発売：プログラミングクラブネットワーク）
・原稿についてはICHIGOJAM開発者**福野 泰介様**のブログを参照させて頂いてます。

2023.9.18 Ver.3.05

小樽別院　寺子屋教室
主催　小樽青少年の科学の芽を育てる会
協力　浄土真宗本願寺派 本願寺小樽別院
協力　NPO法人NEXTDAY

IchigoJamを動かそう！

・IchigoJamパソコンの部品配置と名前を覚えよう

液晶モニター

PS/2キーボード

キーボード端子

（変換アダプター）

ビデオ端子

タクトスイッチ

LED

CPU

電源端子

マイクロUSB
5V

電源スイッチ

入　　切

I/O
端子

I/O
端子

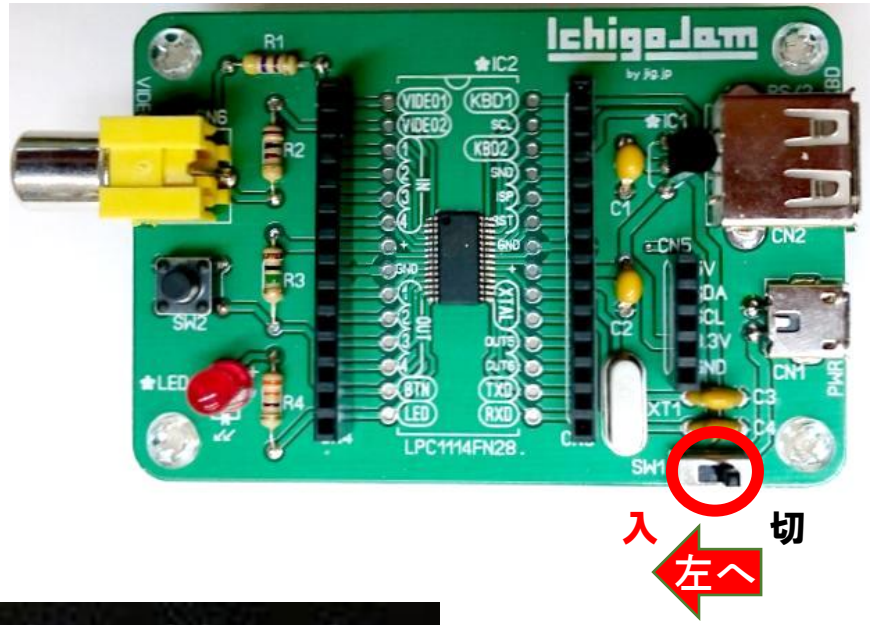
◆電源スイッチを入れよう

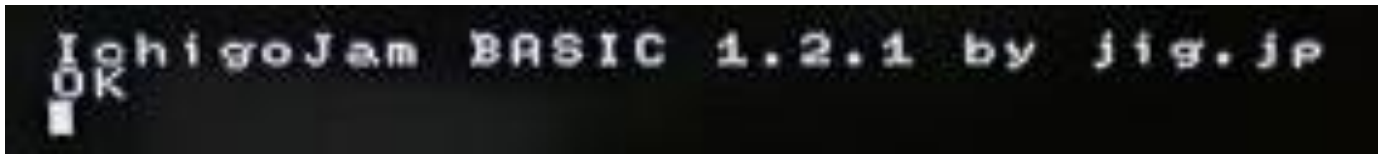
a. 電源スイッチを　右から左　へ動かす

左へ

b. 起動すると画面に文字が表示されます

ピッ！　と起動音が鳴ります。





命令（コマンド）が正しく動作したときに表示されます。

IchigiJamから「わかったよ！」「動いたよ！」と話して（表示して）います。

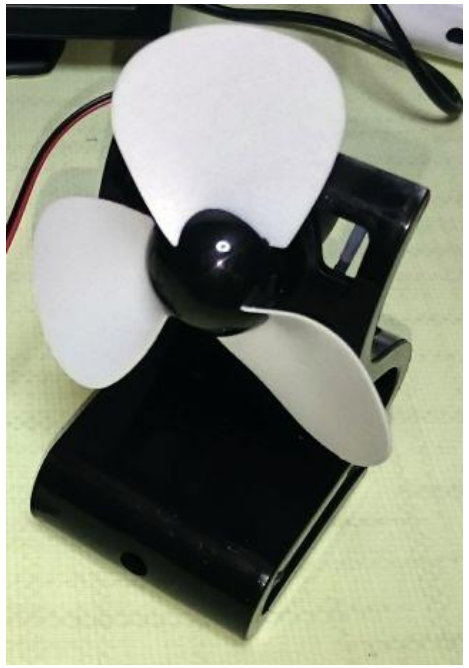
シンタックス　エラー

「分からないよ！！」時

きょうのミッション（指令）

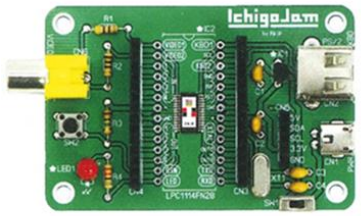
Mission

せんぷう機 を便利にまわそう ！



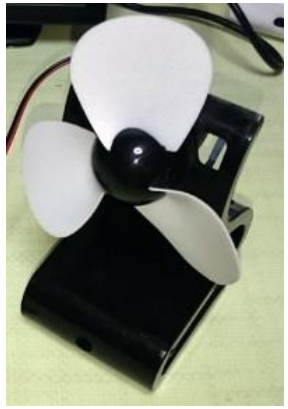
80分

1. せんぷう機を動かしてみよう
2. どんな機能があると便利かなぁ
3. 動き方やと機能を考えよう
   - ・電源スイッチ
   - ・風量調節
4. プログラムで　スイッチ/ボタン　をつくろう
5. 多機能せんぷう機に変身！！



IchigoJam

せんぷう機を便利に動かそう ！



IchigoJamの電気ではモーターを直接まわすことはできません。
そこでトランジスタの仲間の　MOSFET　を使かい、
乾電池からの電気の量を調節してモーターをまわします。

パワーMOSFET　2SK4017 を使い
DCモーターに流す電源スイッチの役割りをプログラムします。

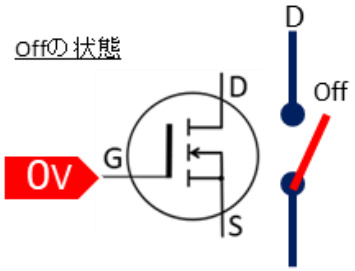
このMOSFETは、力のあるモーターを回したり、たくさんのLED点滅などの大きな電力を必要とするときに利用され、低電圧でも効率よく動かすことがます。

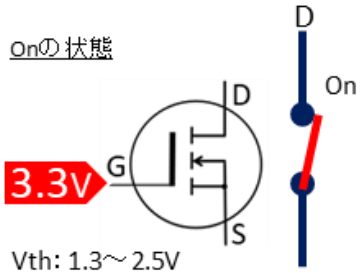
パワーMOSFET

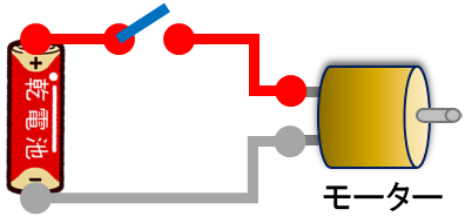
・電源スイッチのOn/Offの仕組み

Offの状態　OUT2から出力されない（0V）時、D-S間のスイッチは切られています。

Onの状態　OUT2から出力された（3.3V）時、D-S間のスイッチはつながります。







このONとOFFを高速で切り替えると、
流れる電気の量を調節できます。
この切替はプログラムで行ないます。

Step１　モーターのまわそう！

コンピュータで回転を制御（せいぎょ）するプログラムつくります

せんぷう機をつなごう

せんぷう機の配線

CN5の**GND**
一番下の端子

OUT2

コマンドで制御

```
OUT2,1
```
モーターが回転

```
OUT2,0
```
モーターが停止

```
PWM2,1000
```

```
PWM2,0
```

この数値を
500～2000までの範囲で
変えて見よう！

Step２　せんぷう機にはどんな機能があるかな？

身近なせんぷう機を見て確認します。

・電源、運転
・風量（弱/中/強）
・リズム
・タイマー
・首ふり

こうした機能をプログラムにする場合、
動作や動き方を細かく分解して考えます。

■電源の場合

・ボタンを押さない　監 判　⇒　ライトを消す　⇒　電気を切る
・ボタンを押す　　視 断　⇒　ライトが点く　⇒　電気を流す

■風量（弱/中/強）の場合

・弱　　⇒　モーターがゆっくりまわる　⇒　電気の流れる量が少ない

PWM2,800

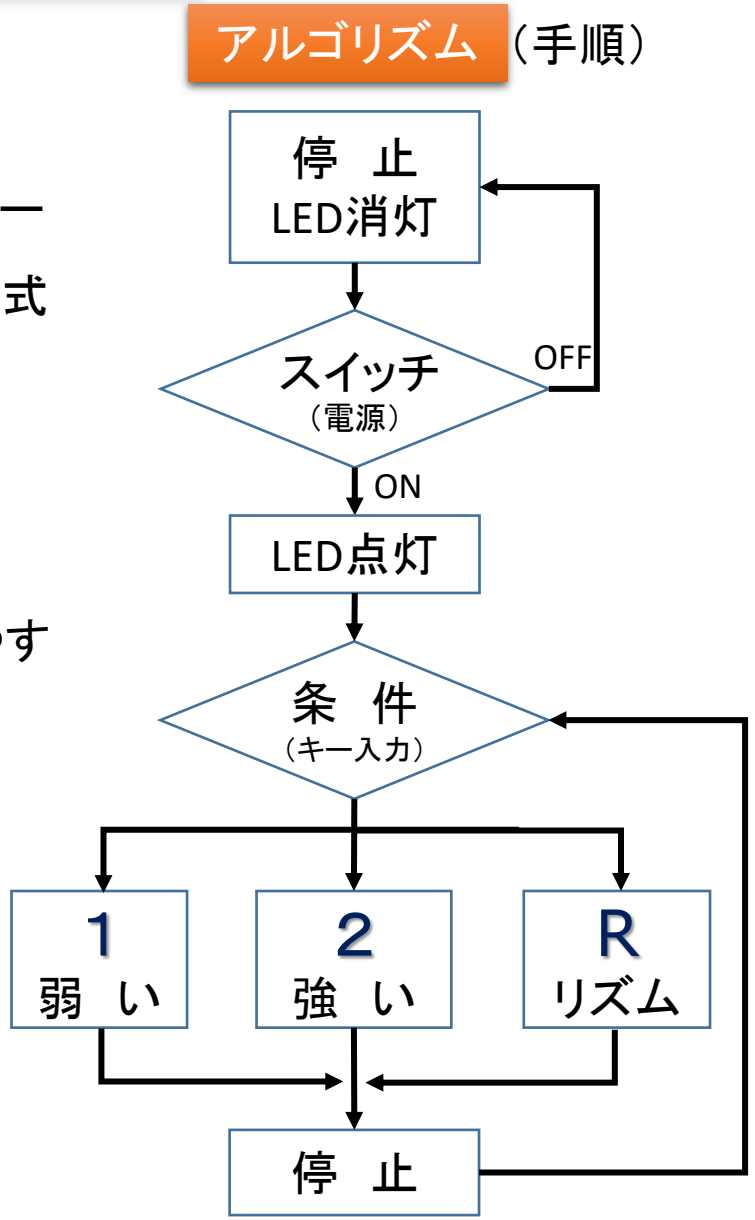
・強　　⇒　モーターが早くまわる　　⇒　電気の流れる量が多い

PWM2,1800

Step３　せんぷう機の機能と動き方を決めよう！

プログラムをつくるときは、
「何を」「どの順番で」「どのように動くのか」をコンピューターに分かるように書く必要があります。それらを分かりやすく図式にしたのを　アルゴリズム　といいます。
細かな動作や働きを分解することで、どのような命令（コマンド）や機器が必要かがわかります。
思いどおりに動かない時やコマンドの間違いなどを見つけやすくなります。

今回は、
・電源ボタン　・風量（弱/強/リズム）　の2つの機能を
プログラムで動かすことに挑戦します。



・電源ボタン 機能をつくる

Step４　センサーを付けよう

センサースイッチ　を端子に接続します。
センサースイッチがどのように反応するか、
次のコマンドでコンピューターの想いを見てみよう！

センサースイッチ

センサースイッチを

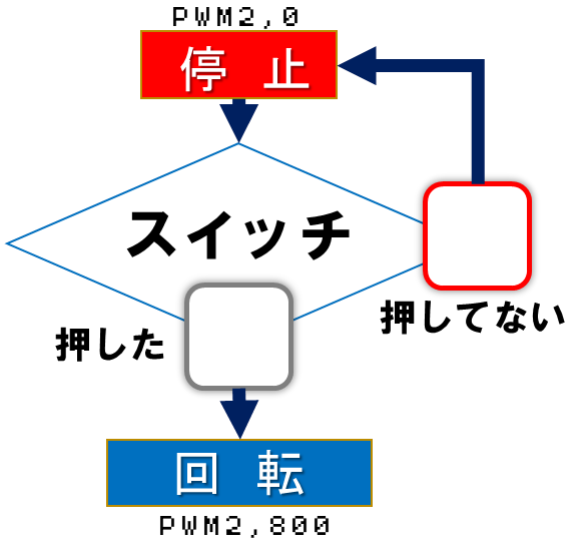
- **押しながら**　？IN(2)　↵　→
- **押さないで**　？IN(2)　↵　→

モニターの数値

Step５　プログラムでまわそう

```
25  LED  1
40 PWM2,1000:WAIT60
50 PWM2,0:WAIT30
60 LED 0
```

```
LOAD 1 ↵
LIST ↵↵
```



◆センサースイッチの数値（0か1）をコンピューターに判断させて
モーターを回すようにします。

もし　●●●　ならば　□□□　する
ＩＦ　●●●　ＴＨＥＮ　□□□

●●●には条件を書きます
□□□には動作を書きます

もし　IF　IN(2)＝0
スイッチが
押していなければ

ならば　THEN　PWM2,0
停止する

20行　と　90行　にプログラムを追加しよう！

電源スイッチ

追加 **20 IF IN(2)=0 THEN PWM2,0:GOTO20**

```
25  LED  1
40  PWM2,1000:WAIT60
50  PWM2,0:WAIT30
60  LED 0
```

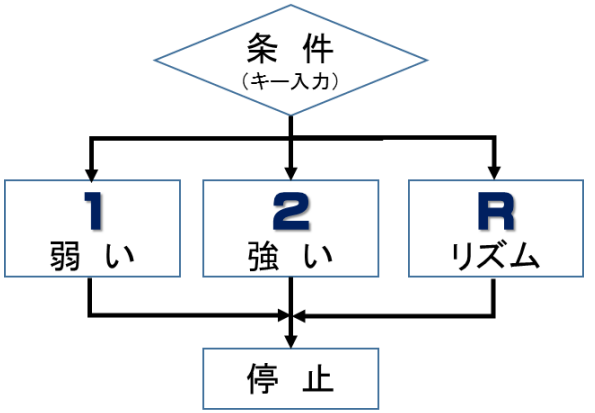
追加 **90 GOTO 20**

プログラムを
修正したら　エンターキー

プログラムを
追加・修正・さくじょ した時は
必ず ＬＩＳＴ コマンドで確認！

・風量（弱/強/リズム） 機能をつくる

キーボードの一つ一つに 数値 が割りあてられています。
その数値を キーコード といいます。
そのキーコードに 風量（弱/強/リズム） とを関連付けることで
選択ボタンとして機能させます。




```
20 IF IN(2)=0 THEN PWM2,0:GOTO20
25  LED  1
```

追加

```
30  A=INKEY() : ? A
```

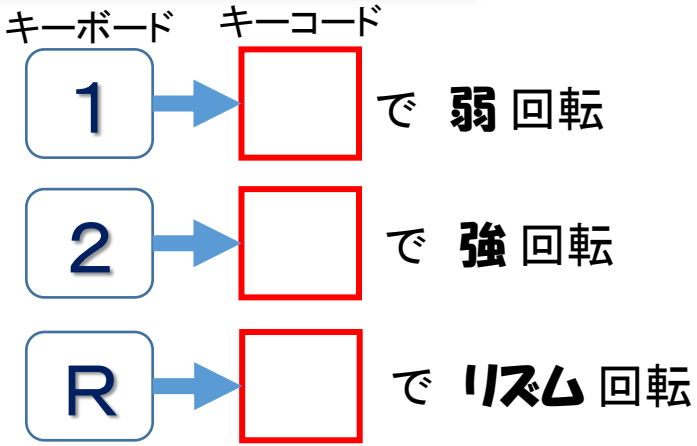
```
40 PWM2,1000:WAIT60
50 PWM2,0:WAIT30
60 LED 0
90  GOTO 20
```

Step６　キーコードで風量を選べるようにしよう！

プログラムを実行して、

・数字の １ と ２
・英字の Ｒ

のキーコードを調べよう。



Step７　プログラムを完成させよう！

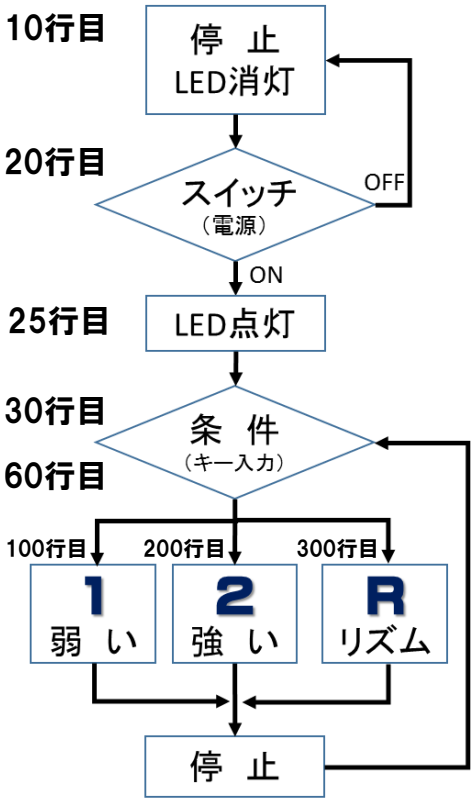
プログラムを **修正**したら ↵ エンターキー

プログラムを **追加・修正・さくじょ** した時は 必ず ＬＩＳＴ コマンドで確認！

```
LOAD 3
LIST
```

3番のプログラムを読み込みます

キーコードを使いどのキーが押されたかを
プログラムで判断させて動かします。




```
10 PWM2,0:LED0                          ← プログラムが動いた時の、一番初めの状態
20 IF IN(2)=0 THEN PWM2,0:GOTO20
25 LED1:WAIT2:BEEP 30,60                ← 電源スイッチがONになるとLED点灯
30 A=INKEY() : ?A                       ← キーボードからのキーコードを取得
40 IF A=[ ] THEN GOSUB 100              キーコードによって実行する
50 IF A=[ ] THEN GOSUB 200              プログラムに移動
60 IF A=[ ] THEN GOSUB 300
90 GOTO 30
100 PWM2,800:WAIT300
110  PWM2, 0:BEEP:RETURN
200 PWM2,1800:WAIT300
210  PWM2, 0:BEEP:RETURN
300 FOR I=1 TO 10          ← 10回繰り返す
310  B=RND(10)             ← 0～9までの乱数がでる
320 PWM2,500+B*80:BEEP:WAIT30
330 NEXT                   ↑乱数によって電気の量が自動的に変化する計算式
340  PWM2, 0:BEEP:RETURN
```

1 弱い　2 強い　R リズム

Step8　プログラムをどんどん改造しよう！

※終了処理を考えて追加してみましょう。

スイッチをもう一度押したらLEDを消して
プログラムを終了します。

※いろいろ音色を変えてみましょう。

BEEPコマンドの値を変更する！

※それぞれの速度を変えてみましょう。

PWMコマンドの値を変更する。

※それぞれの時間を変えてみましょう。

WAITコマンドの値を変更する。

自分のオリジナル改造をしてみましょう。

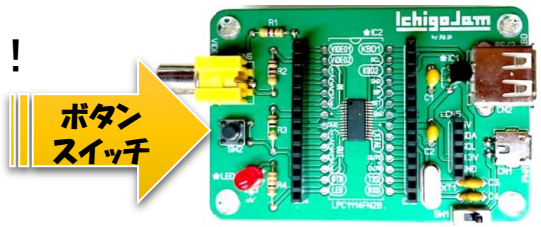
解らない時は講師の先生に聞いてみましょう。

さまざまなプログラムに挑戦！！

※新しいプログラムを入力するときは、
コマン NEW で
コンピューターの記憶を消してから！

```
10 ?BTN()
20 LED BTN()
30 GOTO 10
```

IchigoJamのボタンスイッチを押すと！




```
10 CLS
20 FOR I=1 TO 10
30 A=I+1:?A
30 GOTO 20
```

TO 10 を
TO 100 や TO 1000 に変更してみよう！

インスタントラーメンタイマー

```
10 N=3:C=60:CLT:CLS:LED 0
20 T=N*60*C-TICK()
30 IF T<=0 LED 1:END
40 S=T/C:M=S/60:S=S%60
50 LC 0,0:? M;"m";S;"s ":GOTO 20
```

どんなメロディが聞こえるかなぁ？

```
10 PLAY"O4L8 E2E4D4E4ED>B2<F4FFA4FEFEDDE2"
```

コマンドをおぼえよう　！！

プログラムをリストするには

LIST ↵

範囲を指定してリスト
LIST　10,100

プログラムを実行するには

RUN ↵

プログラムをメモリーに記憶させるには

SAVE ↵

(0～3)の4個のメモリーがあります

プログラムをメモリーから読みだすには

LOAD ↵

メモリーの内容は電源を切っても記憶しています

メモリーの内容をみるには

FILES ↵

プログラムをすべて削除するには

NEW ↵

画面表示をすべて消します（クリア スクリーン）

CLS ↵

プログラムの行削除は行番号だけを打ちます

10 ↵

実行しているプログラムを停止するには

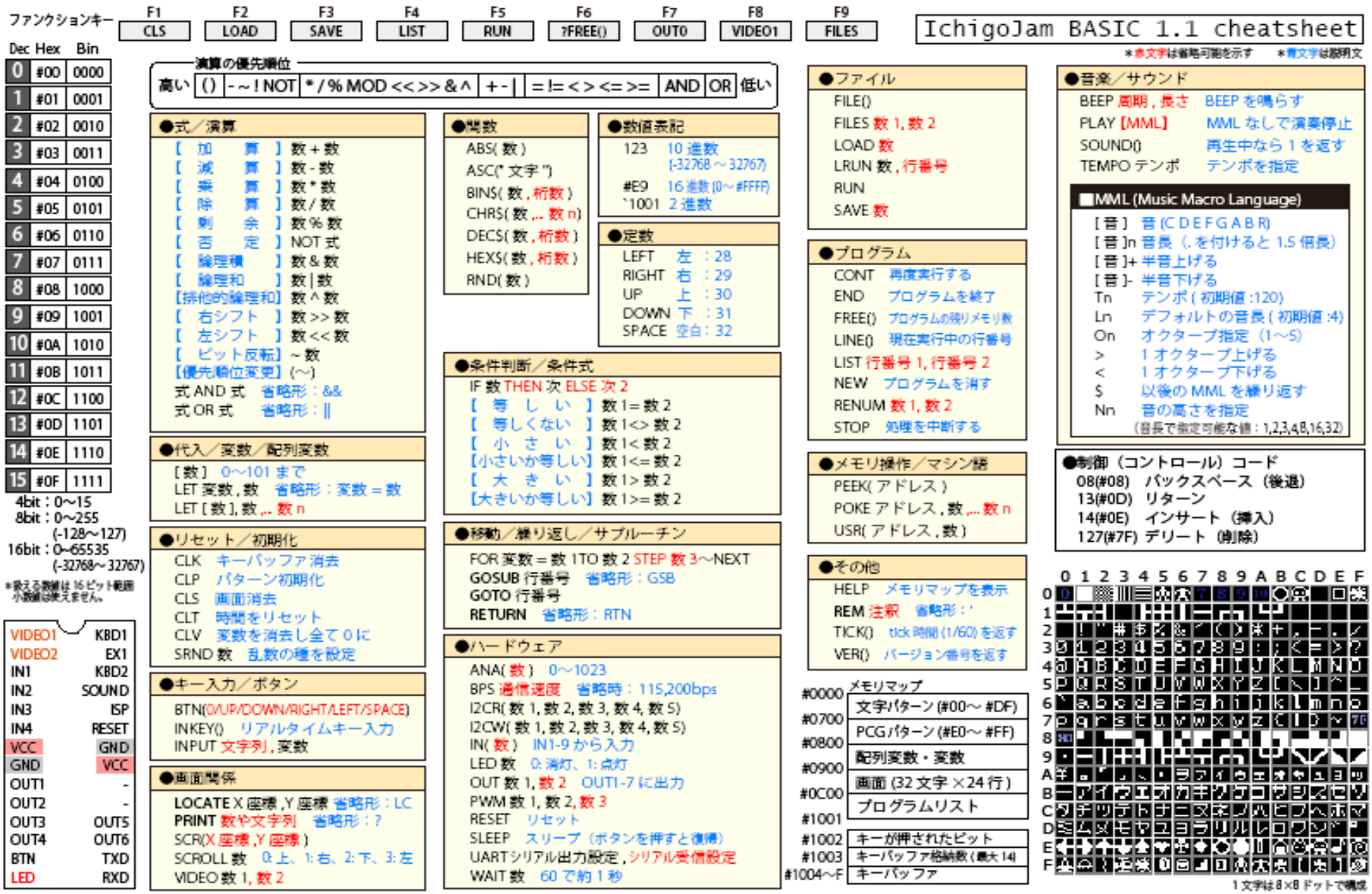
Esc キーを押します

エスケープと読みます
(Escape：逃れる、抜け出す、脱出する)の省略型です

コマンドはキー入力だけでなくファンクションキーでも代用できるよ

F1	F2	F3	F4	F5	F6	F7	F8	F9
CLS	LOAD	SAVE	LIST	RUN	?FREE()	OUT0	VIDEO1	FILES

「IchigoJam BASIC 1.4 コマンド一覧」https://fukuno.jig.jp/app/csv/ichigojam-cmd.html



「IchigoJam 1.2 チートシート　by OpenSpace」

http://www.openspc2.org/reibun/IchigoJam/etc/cheat-sheet/0002/