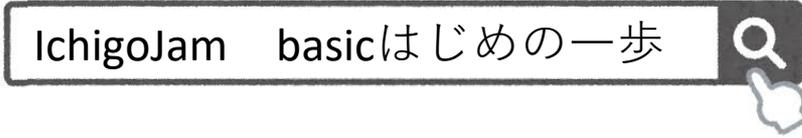




<https://15jamrecipe.jimdofree.com/basic> はじめの一步/



◆ IchigoJam BASIC 1.4 コマンド一覧

<https://fukuno.jig.jp/app/csv/ichigojam-cmd.html>



■ IchigoJam プリント

<https://ichigojam.github.io/print/ja/>

数を きおく



PRINT A

? A

A = 10

? A

きおく して計算

A = 10

B = 4

? A + B

計算記号

? A - B

? A * B

? A / B

? A % B

引き算

かけ算

わり算

わり算の余り

※ A や B の値を変えてみよう!

1 行入力した後は Enter (return) キーを押す

LEDを点ける/消す！ **LED**

```
LED 1 ⌵  
LED 0 ⌵
```

音を出そう！ **BEEP**

```
BEEP ⌵  
BEEP 100, 60 ⌵  
A=30 ⌵  
BEEP 100, A ⌵
```

BEEP (音程 時間
数1 , 数2)
1~255 1~(60で1秒)

おとがくをつくるPLAY (プレイ)！ **PLAY**

C D E F G A B
ど れ み ぶ あ そ ら し

```
PLAY "CDE" ⌵  
PLAY "04 CDEFGAB 05 C" ⌵
```

o (オー) はオクターブ

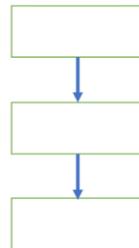
数字や文字 を表示しよう！

```
PRINT "ABC" ⌵  
? "1+2=" ⌵
```

表示する文字を
" " (ダブルクォーテーション) で囲みます
「SHIFT (シフト) を押したまま数字の2キーを押す」

LEDの点滅を プログラミングしよう！

順次



```
10 LED1:WAIT60
```



```
20 LED0:WAIT60
```



ラン
RUN



F5キー (プログラム実行)

- ・ 10回点滅は F5を10回
- ・ 100回点滅は F5を100回

もっと便利なプログラムに改良！

```
10 LED1:WAIT60
```



CLS

画面表示をすべて消す
(クリアスクリーン)

```
20 LED0:WAIT60
```



うえいと

WAIT

数値だけ待つ
(60で1秒)

ころん

:

コマンドを後ろに続けて
書くときの区切り記号

ごーつー

GOTO

指定した行番号へ飛ぶ

追加

```
30 GOTO 10
```



リスト

LIST



(プログラム表示)

F4キー



実行停止
(エスケープキー)

ラン

RUN



F5キー

プログラムを

修正したら



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！

NEW

プログラムを全部消す

次のプログラムを入力して実行しよう。

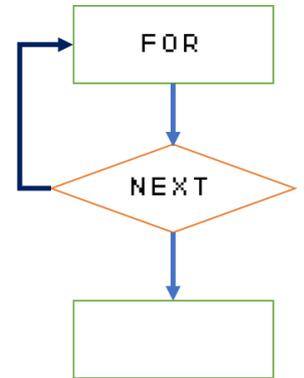
```
10 FOR I=1 TO 10
20 PRINT I
30 NEXT
```



- ・モニターに表示される I の値はどのように変化しましたか？
- ・指定した回数を繰り返します。
- ・この繰り返しの仕組みを使って、1 から 10 までを足した時の計算式を作りましょう。
20 行目を改造します。

次のプログラムはどのように動作するか考えてみよう。

```
10 FOR I=1 TO 10
20 LED I%2:WAIT 30
30 NEXT
```



・押しボタンセンサーでLEDを点滅させよう

しーえるえす

CLS

F1

ぷりんと

PRINT

?

画面を全部消す

数や文字列を表示する
文字列は“で囲む
;で連結できる

タクトスイッチ BTN() の状態を画面で観察

```
10 PRINT BTN()  
20 GOTO 10
```

> 押ししていると . . .



> 押しなければ . . .



BTN() の値をLEDの点滅に利用

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

変数 アルファベット 1文字 A~Z が変数で使え、数値や文字を入れることができる。

```
A=1:PRINT A
```

```
LET B,2:PRINT B
```

```
A=5:? A
```

変数に文字列を入れる場合は、

文字列は“” (ダブルクォーテーション) で囲み、
行番号を付けて書く。
STR\$()で取り出す。

```
10 A="HELLO"
```

```
PRINT STR$(A)
```

・簡単な電卓プログラム

```
10 INPUT A
20 INPUT B
30 C=A+B
40 PRINT C
50 END
```

・配列を使った計算プログラム

5つの配列（[1]～[5]）に数値を記録して、その合計を計算

```
10 CLV
20 FOR I=1 TO 5
40 INPUT [I]
50 WAIT 60*1
60 NEXT
70 FOR J=1 TO 5
80 A=[J]+A
90 NEXT
100 PRINT A
```

30 ?"GETKION"

を加えると気温データを記録できる。

このプログラムを工夫して、合計だけでなく、平均や最大や最小も計算できる。

配列

BASICでは[0]～[101]の102の配列（箱）が使えます。

変数同様に数値を入れられます。

入れられる値は変数同様に-32768～32767の整数です。

```
LET [2],2:PRINT [2]
```

配列[2]に2を入れる

```
[100]=55:PRINT [100]
```

配列[100]に55を入れる

```
LET [0],1,2,3,4,5,6
```

配列[0]から1～6までの数字を入れる

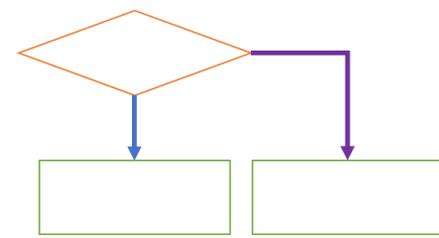
```
PRINT[2]
```

```
PRINT[4]
```

条件分岐

条件によって実行する内容を変える

「もし〇〇の時は□□をする」



「もしボタンの値が0の時は10行目に行く（10行目の命令を繰り返す）」

```
10  IF  BTN( )=0  GOTO  10
20  BEEP
50  GOTO  10
```

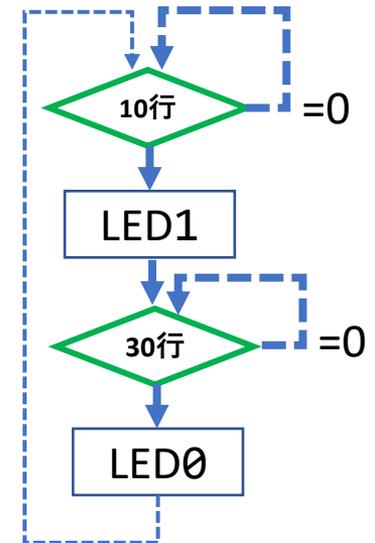
行番号に代わってラベルを使用することができます @ラベル名

```
10  @BTN
15  IF  BTN( )=0  GOTO  @BTN
20  BEEP
50  GOTO  @BTN
```

ボタンを1回おすたびにLEDを点けたり消したりしてみよう

```
10  IF  BTN( )=0  GOTO  10
20  LED1
30  IF  BTN( )=0  GOTO  30
40  LED0
50  GOTO  10
```

引用 <https://ichigojam.github.io/print/ja/IF.html>



LEDが点滅してるかな？

どこを修正すると点いたり消したりできる考えよう！

繰り返した回数 (時間) を表示

一定間隔で繰り返すプログラムで、その回数を表示する

1 秒間隔でLEDの点滅を繰り返すプログラム

```
30 LED1:WAIT30
40 LED0:WAIT30
60 GOTO 20
```

このプログラムに繰り返した回数を表示するプログラムを追加

```
10 CLV:I=1
20 PRINT I
30 LED1:WAIT30
40 LED0:WAIT30
50 I=I+1
60 GOTO 20
```

CLV

変数や配列の値を全てクリア(初期化)する

繰り返した時間 (秒) を表示するプログラムを追加

```
10 CLT
30 LED1:WAIT30
40 LED0:WAIT30
50 PRINT TICK()/60
60 GOTO 30
```

CLT

時間のカウンタを0にリセットする。

TICK()

IchigoJamをONにした時 (またはCLTで時間をリセット) からどれだけ時間がたったかを数字で返す。数は60で大体1秒くらい。32767 (約9分) まで数えたら、また0に戻って数え始める。

IchigoJam (SkeyberryJAM) の電源を切ると、入力したプログラムは消えてしまいます。そこでプログラムを保存したり読み出すことができます。

保存領域は、

- ・ IchigoJam : 4 (本体内)
- ・ SkeyberryJam : 15 (外部記憶装置EEPROM)

```
FILES
0 ?"APC
1 'THINGS BORD
2
3
4
5
6
7
8
9
10
11
12
13
14
```

◆保存プログラムの確認

FILES (ふぁいるず) 命令で確認します。

保存領域の番号と、プログラム 1 行目の内容が表示されます。

保存プログラムの確認

```
FILES ↵
```

◆プログラムの保存

SAVE (せーぶ) 命令を使います。

SAVE 領域番号

3番に保存

```
SAVE3 ↵
```

◆プログラムの保存

LOAD (ロード) 命令を使います。

LOAD 領域番号

3番を読み出し

```
LOAD3 ↵
```

▼保存プログラムを直接読みだして実行

LRUN命令を使うと、保存しているプログラムを直接実行できます。

5番を読み出して実行

```
LRUN 5 ↵
```

5番を読み出して、
100行目から実行

```
LRUN 5,100 ↵
```

コンピュータの



プログラムとは、

コンピュータが計算（動作）するための手順書

その手順書には、主に3つの仕組みがあり、**アルゴリズム**とといいます。

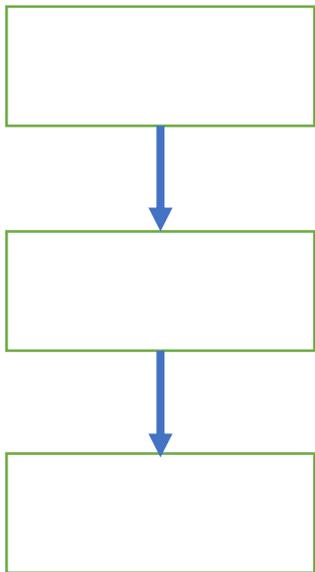
- **順次** : 先頭から順番に処理を実行
- **条件分岐** : 条件によって実行する内容を変える
- **繰り返し** : 指定した条件を繰り返して実行

フローチャート

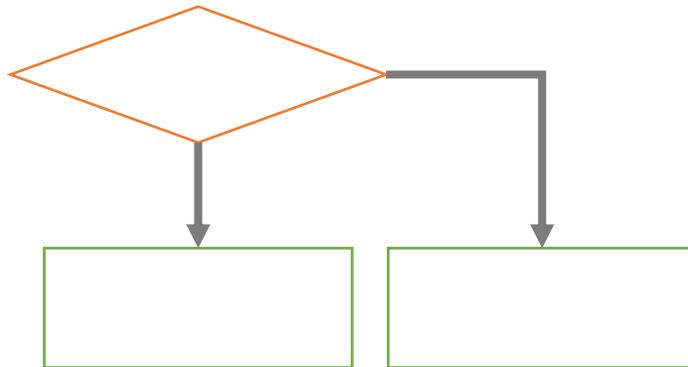
プログラム（手順書）の処理の流れをフローチャートとといいます。

- ・ 図や手順を図にすることで、文章より分かりやすい
- ・ 見ただけで全体の構成が分かりやすい
- ・ プログラムを作りやすい

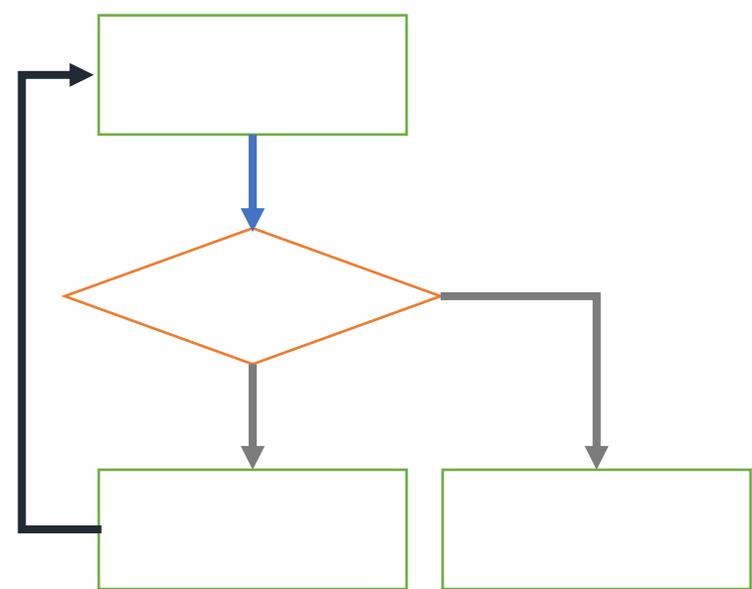
順次



条件分岐

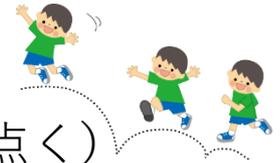


繰り返し



身近なモノや動作で、この3分類を探してみよう！

ちやれんじ-1!



● 電気が流れる (LEDが点く)

OUT端子番号 コマンド	6	5	4	3	2	1	2進表示	10進表示
OUT 0	○	○	○	○	○	○	000000	0
OUT 1	○	○	○	○	○	●	000001	1
OUT 2	○	○	○	○	●	○	000010	2
OUT 3	○	○	○	○	●	●	000011	2 + 1
OUT 4	○	○	○	●	○	○	000100	4
OUT 5								4 + 1
OUT 6								4 + 2
OUT 7	○	○	○	●	●	●	000111	4 + 2 + 1
OUT 8								8

OUT ` 000001 ↩

バッククォー | Shiftキー
をおしながら

