

IchigoJam はじめのいっぽ

LEDをひからせよう

LED1

LED1、と、おして「enter」キー

エンター

LEDをけそう

LED0

ぎょうのおわりにはエンターキー

コンピューターに「まで = WAIT (ウェイト)」

WAIT120

WAIT120で2びょうまつ、では60だと？

WAIT60

LEDを1びょうひからせる (**:** コロンでつなく)

LED1:WAIT60:LED0

カーソルキーの上キーを2かいおす

右キーを10かいおして、0のばしょまでうごかす

Backspace (バックスペース) キーで6をけす

12とうち、さいごにエンターキー

LED1:WAIT120:LED0

LEDをてんめつさせよう

(くうはく = スペースキー、まんなかのながいキー)

```
10 LED1:WAIT60
20 LED0:WAIT60
30 GOTO10
RUN
```

左上のESC (エスケープ) キーでとまるよ
RUNのかわりに F5キーをおしてみよう

プログラムをかいぞうしよう

LIST

リスト、F4キーでもOK!

はやくてんめつさせるにはどこをかえる？
かえたらンターキーをわすれずに！

つくったプログラムはほぞんしよう (0~3の4つ)

SAVE0

セーブ、F3キーでもOK!

スイッチをきっても、もとどおり

LOAD0

ロード、F2キーでもOK!

つぎのプログラムをはじめるまえに

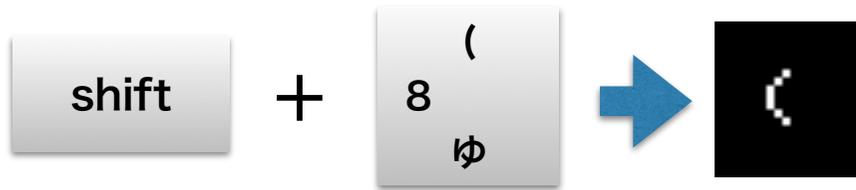
NEW

ニュー



IchigoJam ミニゲームズ

 () や”はシフトキーをおしながらおす！



はんのうそくどゲーム

(LEDひかったらすぐ、IchigoJamほんたいボタン)

```
10 LED0
20 WAIT RND(180)+60
30 LED1:CLT
40 IF BTN()=0 GOTO 40
50 ?TICK()/60
```

かわくだりゲーム (カーソル左右でよける！)

```
10 X=16:CLS:CLT
20 K=INKEY()
30 X=X-(K=28)+(K=29)
40 IF SCR(X,5) END
50 LC X,5:"0"
60 LC RND(32),23:"*"
70 WAIT 3
80 GOTO 20
```

こうそくタイピングゲーム

(AからZまですばやくおせ)

```
10 N=65:CLT
20 ?CHR$(N);
30 IF INKEY()<>N GOTO 30
40 N=N+1:IF N<>91 GOTO 20
50 ?:"TICK()/60
```

たしざんめいじんゲーム

(かずをうちこんでエンターでこたえる)

```
10 N=0:CLT
20 A=RND(10)
30 B=RND(10)
40 ?A;"+";B;"=";:INPUT C
50 IF C!=A+B ?"NG!":END
60 N=N+1:IF N<10 GOTO 20
70 ?TICK()/60
```

やきゅうゲーム (タイミングよくボタンおす)

```
10 Y=0
20 CLS
30 LC 4,15:"%"
40 LC 5,Y:"0"
50 IF BTN() GOTO 90
60 Y=Y+1
70 WAIT 6
80 GOTO 20
90 IF Y=15 ?"HIT!"
```

スクリーンジャック (キーボードでうつと?)

```
10 C=1:CLS
20 LC RND(32),RND(22)
30 ?CHR$(C)
40 K=INKEY():IF K=C
50 GOTO 20
```



BASICでプログラミング!

こどもパソコン **IchigoJam**



<http://ichigojam.net/>

IoT はじめのいっぽ

音を鳴らす

サウンダー

右側のピンソケットの上から4つ目と上から7つ目に差し込む
BEEP . . . ピツとなる
PLAY"CDE" . . . ドレミ
PLAY" CDE2" 2や8で2分音符、8分音符など
PLAY" CRC" Rで休符
PLAY" O4C05C" オクターブをかえる0
PLAY" T240CDE" テンポをかえるT
PLAY" EDC\$CDE" くりかえし記号の\$
PLAY . . . PLAY だけかくと音楽停止

サーボ

ジャンパー線で3本接続

サーボの茶色-IchigoJam の GND
サーボの赤色-IchigoJam の 5V
サーボのオレンジ色-IchigoJam の OUT4 (左側下から3つ目)

PWM4,135 . . . センター
PWM4,135-85 . . . 右いっぱい回転
PWM4,135+85 . . . 左いっぱい回転

ボタン #103

IN1につなぐ

?IN (1)
0 . . . 押されていない
1 . . . 押されている

IN2につなぐ

?IN (2)
0 . . . 押されていない
1 . . . 押されている

角度センサー (ポリューム) #104

IN2につなぐ

?ANA (2)
0 . . . 左回転いっぱい
1023 . . . 右回転いっぱい

光センサー #109

IN2につなぐ

?ANA (2)
0 . . . とっても明るい
1023 . . . 真っ暗

温度センサー #108

IN2につなぐ

?ANA (2)
250 常温ぐらい、大きいと温かい、小さいと冷たい

LED #101

IN4につなぐ

OUT11,1 . . . LED をつける
OUT11,0 . . . LED を消す

ネットにつなぐ (さくらのIoT Platform α)

C= (せきばんごう)
N= (おくりたい数)
POKE #700,48,C,73,N,N>>8,0,0:?I2CW (79,#700,1,#701,6)

IchigoJam BASIC リファレンス ver 1.2

キーボード操作

操作	解説
キー	文字を入力する
Shift / シフト	キーと共に押し記号や小文字などを入力する
カタカナ	アルファベットとカタカナ（ローマ字入力）を切り替える（右Alt、CTRL+SHIFT / コントロール+シフトでも可）
Enter / エンター	コマンドを実行する（プログラム変更時その行でEnterキー）
Shift+Enter / シフト+エンター	行を分割する
ESC / エスケープ	プログラムの実行、リスト表示、ファイル一覧表示を止める
カーソルキー	カーソルキーを移動する
Backspace / バックスペース	カーソルの前の文字を消す
Delete / デリート	カーソルにある文字を消す
左ALT / オルト	0-9/A-Kと合わせて押すことで拡張文字入力（SHIFT押しながら切り替え）、[と合わせて押して]、]と合わせて押して]や¥の入力
Home End / ホーム エンド	カーソルを行頭へ移動、カーソルを行末へ移動
Page Up Page Down / ページアップ ページダウン	カーソルを画面上へ移動、カーソルを画面下へ移動
Caps / キャプス	大文字と小文字を切り替える
Insert / インサート	キーボードの上書きモード/挿入モードを切り替える（CTRL+ALTでも可能）
ファンクションキー	F1:画面クリア、F2:LOAD、F3:SAVE、F4:LIST、F5:RUN、F6:FREETO、F7:OUTO、F8:VIDEO1、F9:FILES
ボタン	押しながら起動でFILEOを自動実行する

初級コマンド

コマンド	解説	例
LED 数 / エルイーディー	数が1なら光り、0なら消える	LED 1
WAIT 数1[.数2] / ウェイト	数1の数値フレーム分待つ 60で約1秒。省略可の数2指定で低電力化、数1のマイナス指定で走査線分待つ(261でWAIT1と同等)	WAIT 60
:/ コロ	コマンドを連結する	WAIT 60LED 1
行番号 コマンド	プログラムとしてコマンドを記録する	10 LED1
行番号	指定した行番号のプログラムを消す	10
RUN / ラン	プログラムを実行する [F5]	RUN
LIST (行番号1[.行番号2]) / リスト	プログラムを表示する [F4]（行番号1で1行表示、行番号1がマイナスでその行まで表示、行番号2指定でその行まで表示、行番号2が0の継続わりまで表示、ESCで途中停止）	LIST 10,300
GOTO 行番号 / ゴートウ	指定した行番号へ飛ぶ(試み指定不可)	GOTO 10
END / エンド	プログラムを終了する	END
IF 数 [THEN] 次1 [ELSE 次2] / イフ・ゼン・エルス	数が0でなければ次1を実行し、0であれば次2を実行する（THEN,ELSE以降は省略可）	IF BTN0 END
BTN(数) / ボタン	ボタンが押されているか、そうで無いとき0を返す（数：0付属ボタン)/UP/DOWN/RIGHT/LEFT/SPACE、省略で0	LED BTN0
NEW / ニュー	プログラムを全部消す	NEW
PRINT (数や文字列) / プリント	文字を表示する（文字列は"で囲む、"で連結できる）省略形：?	PRINT "HI"
LOCATE 数.数 / ロケート	次に文字を書く位置を横、縦の順に指定する（縦=1で無表示）省略形：LC	LOCATE 3.3
CLS / クリア スクリーン	画面を全部消す	CLS
RND(数) / ランダム	0から数未満の正数をランダムに返す	PRINT RND(6)
SAVE (数) / セーブ	プログラムを保存する（0~3の4つ、100-227 外付けEEPROM、省略で前回使用した数） ボタンを押した状態で起動すると0番を読み込み自動実行	SAVE 1
LOAD (数) / ロード	プログラムを読み出す（0~3の4つ、100-227 外付けEEPROM、省略で前回使用した数）	LOAD
FILES (数1[.数2]) / ファイルズ	数1(省略可)~数2のプログラム一覧を表示する（EEPROM内ファイル表示に対応、0指定ですべて表示、ESCで途中停止）	FILES
BEEP (数1[.数2]) / ビープ	BEEPを鳴らす 周期(1-255)と長さ(1/60秒単位)は省略可 ※SOUND(EX2)-GNDに圧電サウンダーなどの接続必要	BEEP
PLAY (MML) / プレイ	MMLで記述した音楽を再生する MML省略で停止 ※SOUND(EX2)-GNDに圧電サウンダーなどの接続必要（次項のMML参照）	PLAY "\$CDE2CDE2"
TEMPO 数 / テンポ	再生中の音楽のテンポを変更する	TEMPO 1200
数 + 数	足し算する	PRINT 1+1
数 - 数	引き算する	PRINT 2-1
数 * 数	掛け算する	PRINT 7*8
数 / 数	割り算する（小数点は以下切り捨て）	PRINT 9/3
数 % 数	割り算した余りを返す	PRINT 10%3
(数)	カッコ内は優先して計算する	PRINT 1+(1*2)
LET 変数数 / レット	アルファベット1文字を変数としての値を入れる（配列に連続代入可能）省略形：変数=数	LET A,1
INPUT [文字列]変数 / インプット	キーボードやUARTからの入力1文字を変数に入れる（文字列とコマンドは省略可）	INPUT "ANS",A
TICK0 / ティック	CLTからの時間を返す (約1/60秒単位)1進む0	PRINT TICK0
CLT / クリア ティック	時間をリセットする	CLT
INKEY0 / インキー	キーボードやUARTから1文字を入力する（入力がない時は0、UARTから0が入力された時は#100）	PRINT INKEY0
CHR\$(数) / キャラ	文字コードに対応する文字を返す（コマンド区切りで連続表記可）	PRINT CHR\$(65)
ASC(文字) / アスキー	文字にする文字コードを返す	PRINT ASC"A"
SCROLL 数 / スクロール	指定した方向に1キャラクター分スクロールする（0/UP上、1/RIGHT右、2/DOWN下、3/LEFT左）	SCROLL 2
SCR0(数) / スクリーン	画面上の指定した位置に書かれた文字コードを返す（指定なしで現在位置）別名：VPEEK	PRINT SCR(0,0)
数 = 数	比較して等しい時に1、それ以外で0を返す (=でも可)	IF A=B LED 1
数 < 数	比較して等しくない時に1、それ以外で0を返す (!=でも可)	IF A<B LED 1
数 <= 数	比較して以下の時に1、それ以外で0を返す	IF A<=B LED 1
数 <= 数	比較して未満の時に1、それ以外で0を返す	IF A<B LED 1
数 >= 数	比較して以上の時に1、それ以外で0を返す	IF A>=B LED 1
数 > 数	比較してより大きい時に1、それ以外で0を返す	IF A>B LED 1
式 AND 式 / アンド	どちらの式も1の時に1、それ以外で0を返す (&でも可)	IF A=1 AND B=1 LED 1
式 OR 式 / オア	どちらかの式が1の時に1、それ以外で0を返す (でも可)	IF A=1 OR B=1 LED 1
NOT 式 / ノット	式が0の時に1、それ以外で0を返す (!でも可)	IF NOT A=1 LED 1
REM / リマーク	これ以降の命令を実行しない（コメント機能）省略形：'	REM START
FOR 変数=数1 TO 数2 (STEP 数3) NEXT / フォー・トゥー・ステップ・ネクスト	変数に数1を入れ、数2になるまで数3ずつ増やしながらNEXTまでをくりかえす（STEPは省略可、6段まで）	FOR I=0 TO 10:I*NEXT

IN(数) / イン	IN0-10から入力する（0または1）数を省略してまとめて入力できる（IN0,1,4,9はプルアップ、IN5-8,10-11は切り替え時、IN0,9はボタン）	LET A,IN(1)
ANA(数) / アナログ	外部入力端子の電圧(0V-3.3V)を0-1023の数値で返す(2.IN2、5-8.IN5-8(OUT1-4)、0.9.BTN、省略で0)	?ANA()
OUT 数1[.数2] / アウト	外部出力OUT1-11に0または1を出力する 数2を省略してまとめて出力できる（数2に-1指定でINへ切り替え）	OUT 1,1
PWM 数1.数2[.数3] / ビーダブリューム	外部出力OUT2-5に数2で0,0.1msec単位で指定するパルスを出力する（0-2000、周期20msec）、数3で周期を指定（省略時2000=20msec、マイナスイ値指定で周期1/480）	PWM 2,1.00

MML (PLAYコマンド内)

コマンド	解説	例
音	音(C D E F G A B /ドレミファソラシ)を鳴らす（Rは休符、スペースはスキップされる）	CDER FG
音n	長さを指定して音を鳴らす（を付けることと半分の長さ分伸びる）	C4 E2, D1 F32
音+	半音上げる	C+ D+
音-	半音下げる	D- E-
Tn	テンポ (TEMPO命令で後から変更可能) 初期値:120	T96CDE
Ln	長さ指定しないときの長さ(1,2,3,4,8,16,32) 初期値:4	CLBDC
On	オクターブ指定 O1C(低音)からO5B(高音)まで 初期値:3	O3CO2C
<	オクターブ上げる（ver1.1と逆なので注意）	C<C<C
>	オクターブ下げる（ver1.1と逆なので注意）	C>C>C
\$	これ以降のMMLを繰り返す（BGMに便利）	CSDE
Nn	1-255 音の高さ指定してLで指定した長さで鳴らす（BEEP命令と同じ）	N10N5
'	以降のMMLを鳴らさない	CDE

上級コマンド

コマンド	解説	例
CLV / クリア バリイアブル	変数、配列を全部0にする 別名：CLEAR	CLV
CLK / クリア キー	キーバッファとキーの状態をクリアする	CLK
CLO / クリア アウトプット	入力ピンを初期状態に戻す	CLO
ABS(数) / アブソリュート	絶対値を返す（マイナスはプラスになる）	?ABS(-2)
[数]	配列（[0]から[10]までの102コの連続した変数として使えます） [LET][0],1,2,3で連続代入可能	[3]=1
GOSUB 行番号 RETURN / ゴーサブ・リターン	数または式で指定した行番号に飛び、RETURNで戻ってくる 省略形:GSB (30段まで)/RTN	GOSUB 100
DECS(数[.数]) / デシ	数を文字列にします（2番目の数は桁数、省略可）	?DECS(99.3)
#16進数	16進数で数を表示します	?#F
HEXS(数[.数]) / ヘックス	数を16進数の文字列にします（2番目の数は桁数、省略可）	?HEXS(255.2)
?2進数	2進数で数を表示します	?1010
BINS(数[.数]) / バイナリー	数を2進数の文字列にします（2番目の数は桁数、省略可）	?BINS(255.8)
数 & 数	論理積（ビット演算）	?3&1
数 数	論理和（ビット演算）	?3 1
数 ^ 数	排他的論理和（ビット演算）	?A^1
数 >> 数	右シフトする（ビット演算）	?A>>1
数 << 数	左シフトする（ビット演算）	?A<<1
-数	ビット反転（ビット演算）	?~A
STOP / ストップ	プログラムを中断する	STOP
CONT / コンティニュー	実行中の行や、中断した行を再度実行する	CONT
SOUND0 / サウンド	音が再生中なら1、そうで無いとき0を返す	?SOUND0
FREE0 / フリー	プログラムの残りメモリ数を返す	?FREE0
VER0 / バージョン	IchigoJam BASICのバージョン番号を返す	?VER0
LANG0 / ランゲージ	IchigoJam BASICの言語番号を返す（1:日本語、2:モングル語、3:ベトナム語）	?LANG0
RENUM (数1[.数2]) / リナンバー	プログラムの行番号を数1から数2刻みにする（数省略で10、GOTO/GOSUBの飛び先は手で変更必要）	RENUM
LRUN (数) / ロードラン	プログラムの読み込み後、実行します	LRUN 1
FILE0 / ファイル	最後にプログラムの読み込み、書き込みを行った数を返す	?FILE0
LINE0 / ライン	現在実行中の行番号を返す（非実行時は0）	?LINE0
SRND 数 / エスランド	種を指定して乱数を初期化する	SRND 0
HELP / ヘルプ	メモリマップを表示する	HELP
PEEK(数) / ピーク	メモリ読み出し（キャラクターパターン0-#7FFなど）	?PEEK(#700)
POKE 数数 / ポーク	メモリへの書き込み	POKE #700,#FF
COPY 数1.数2.数3 / コピー	メモリコピー 数1のアドレスへ数2のアドレスから数3の長さ分コピー（数3マイナスでコピー方向が逆になる）	COPY #900,0,256
CLP / クリア パターン	キャラクターパターン(#700-#7FF)を初期化します	CLP
"文字列"	文字列の場所を返す	A="ABC"
STR\$(数1[.数2]) / ストリング	文字列を返す（数2(省略可)で長さ指定）	PRINT STR\$(A)
LEN("文字列") / レングス	文字列の長さを返す	PRINT LEN("ABC")
@ラベル / アットマーク	行の先頭に書くとラベルとなり、行番号の代わりとして使える（GOTO @LOOPなど）	@LOOP
VIDEO 数1[.数2] / ビデオ	画面表示非表示を切り替える 0で画面表示を停止し処理高速化（FBで表示）、省略可能な数2でVIDEO0時CPUロックを1/数2に変更し省電力化、数2が2で白黒反転	VIDEO 0
RESET / リセット	IchigoJamをリセットする	RESET
SLEEP / スリープ	プログラムを休止する（ボタンを押すと起動し、LRUNOを実行する）	SLEEP
UART 数1[.数2] / コーアート	数1：シリアル出力設定（0:オフ、1:PRINTのみ、2:PRINT/LC/CLS/SCROLL、3:PRINTのみ/改行コード\n、初期値:2）、数2：シリアル受信設定（0:オフ、1:オン 省略時）	UART 0
BPS 数 / ビーピーエス	シリアル通信速度を変更する(0で初期値の115,200bps、-1:57600bps、-2:38400bps)	BPS 9600
I2CR(数1.数2.数3.数4.数5) / アイツーシー	I2Cで周辺機器から読み込み I2Cアドレス、コマンド送信アドレス・長さ、受信アドレスと長さ	R=I2CR(#50,#700,2,#702,2)
I2CW(数1.数2.数3.数4.数5) / アイツーシー	I2Cで周辺機器に書き込み I2Cアドレス、コマンド送信アドレス・長さ、送信アドレスと長さ	R=I2CW(#50,#700,2,#702,2)
USR(数数) / ユーザー	マシン語呼び出し（注意！高確率でIchigoJamが停止します）	A=USR(#700,0)

CC BY <http://ichigojam.net/>