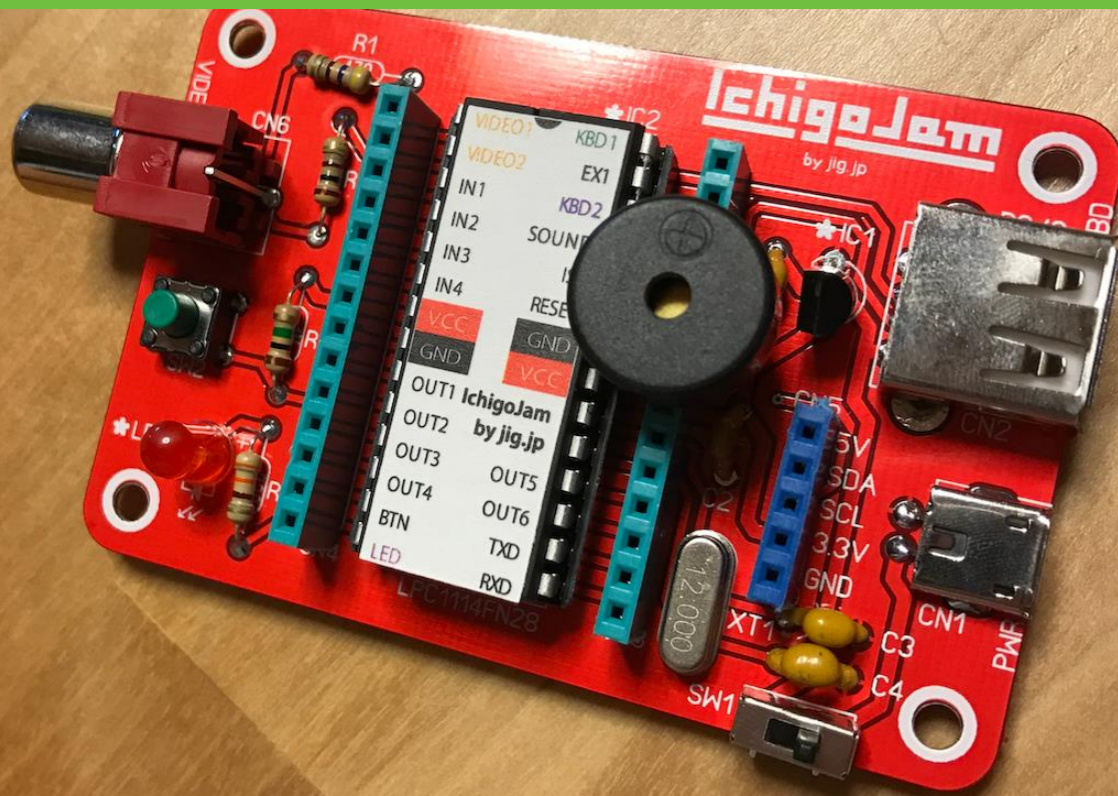


ゲームをつくらう

IchigoJam



ゲーム作りを通して、 プログラムを自分でタイピングして作る

この講座では、次の事柄を、IchigoJamでのゲーム作りを通して体感しながら、学んでいきます。

- ゲーム を 自分でつくる
- 順次処理 を体験し、論理思考を養う
- プログラミング、動作確認 を 何度も繰り返せる
- 指示があつてれば動作。指示が間違っていればエラー表示 or 思わぬ動作
- 自分で作ったプログラムは改造ができる

ゲーム作りを通して、 プログラムを自分でタイピングして作る

- ゲームを自分でつくる
→ 家で遊んでいるゲームは、プログラミングで作れることを体験する
IchigoJamの画面は白黒で、絵も粗いが、楽しさには影響がない
- 順次処理を体験し、論理思考を養う
→ 行の最初にある数字（=行番号）の小さい数字から大きい数字へと
順番に動作が進んでいく。その大きな流れにそって、やりたいことを組
み立てる（=論理思考）
- プログラミング、動作確認を何度も繰り返せる
→ コンピューターの特徴でもある、試行錯誤ができることを体験
思いついた事をコンピューターに伝える方法を、自ら探る
- 指示があつてれば動作。指示が間違っていればエラー表示or思わぬ動作
→ 伝え方の組み立ての検証を、自ら行える
- 自分で作ったプログラムは改造ができる
→ 自ら作ったゲームのルールやキャラクターが変更できることから、つ
くる楽しみ・喜びへつなげる。また、人が作ったプログラムも、自分が
行いたい変更を加える事ができるという事実を知る。

ゲーム作りを通して、 プログラムを自分でタイピングして作る

- スケジュール
 - 講座2a : IchigoJamの準備
 - 講座2b : ゲーム作成 -導入-講座
 - 講座2c : ゲーム作成 -完成-講座
 - 講座2d : ゲーム作成 -改造-
- 機材一覧
 - IchigoJam, キーボード, 電源, テレビ, 各種ケーブル
 - IchigoJamリファレンス or IchigoJamでプログラミング等の参考書
- 指導される方へのメッセージ
 - 数字の変更などによる試行錯誤で、理解が深まります
 - プログラム上の間違いでは、IchigoJamは壊れませんので安心



講座2a : IchigoJamの準備

- 講座1a を参考にして、子供毎にIchigoJamを準備します
- はやく準備が終わった子供は、LED1などの復習
- クイズ3 の答え合わせを行うのも良いです



講座2b：ゲーム作成 -導入-

- スライドが進むごとに、ゲームの要素が動くようになっていきます
- 途中で、プログラミングで使う技術の紹介を少し含んでいます
- 組み込みキャラクターは、シーンに合わせて変更するのも良いです

ゲームをつくらう

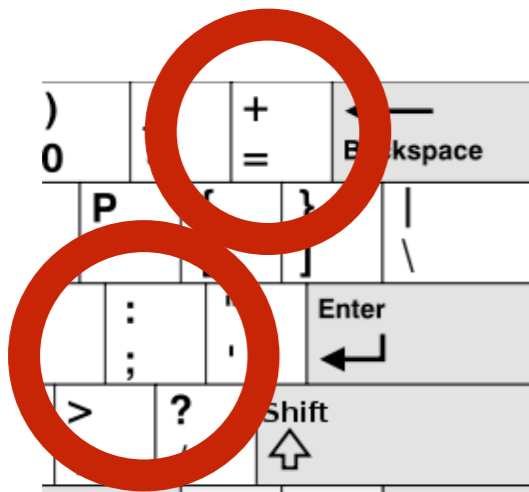


さいしょから（プログラムクリア
）

NEW

ほぞんしたのは
きえないよ

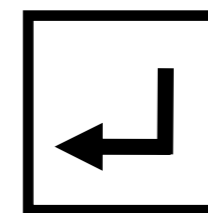




「:」 (コロン) は、キーボード右がわ。
SHIFTと同時に、「;」キーをおす

「=」 (イコール) はキーボード右上。BSとなり。

10 CLS:X=16



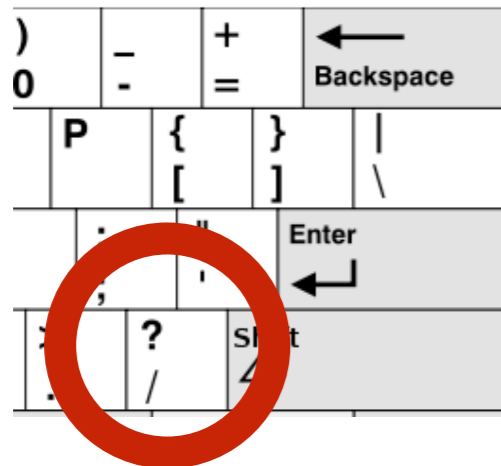
数字で、はじまる行は、すぐに動かない
いで、IchigoJamが、きおくします

メンター：
RUN (もしくはF5キー) で、記憶したプログラムが実行できます

きおくさせよう

NEW, RUN, CLS, ?

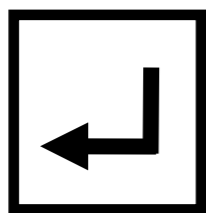
- 命令 : NEW
 - NEW は、現在作成中のプログラムを全部消して、電源を入れたときと同じ状態にする命令です。
- 命令 : RUN
 - RUN は、現在記憶しているプログラムを順番に実行する命令です。
 - ここまでで10行目のみ、入力されて記憶されているので、RUNを実行すると、10行目の内容を、左から順番に実行します。
- 命令 : CLS
 - CLS は、画面に表示されている文字をすべてクリアする命令です。作成中のプログラムは失われません。
- 命令 : ?
 - ? は それに続く文字を画面に表示する命令です。
 - ? は PRINT 命令の省略形です (どちらも同じ動き)
 - ?10 は、数字10を表示。?1+2 は 1+2 の計算結果を表示 (ここでは 3)。
?"ABC" は文字ABCを表示。 ?X は変数Xの内容を表示。



「？」（はてな）は
キーボード右のSHIFT左となり
SHIFTと同時に、おす

はてなマークでがめんにひょうじ

?X



なにがでるかな
?

メンター：
最初のプログラムで箱Xに16を入れました
で?Xをすると、箱Xの中に入っている
情報「数字の16」を表示します。



変数

- プログラミングでの大切な要素に 変数 があります
- 変数とは、数字や文字の入れる物と 考えます。式の左辺の文字（変数）に式
の右辺の数字や文字を入れたら、記号の意味は「代入」して一時的に記憶さ
す。次の書き方は、どちらも同じ意味です。
- LET A,A+1 と A=A+1 ※「=」記号の意味が数学の等価ではなく代入
- 例：変数Aの初期値から相対的に文字を表示する
 - 10 A=10:LOCATE 10,A:PRINT "HELLO"
 - 20 A=A+1:LOCATE 10,A:PRINT "ICHIGO"
 - 30 A=A+1:LOCATE 10,A:PRINT "JAM"
- 上記プログラムの解説：
 - 10 A に 10を入れて、“HELLO” を表示
 - 20 A に A（いまは10） +1を入れて、“ICHIGO” を表示
 - 30 A に A（いまは11） +1 を入れて、“JAM” を表示
- 指導される方へのメッセージ
 - 変数の正確な理解がなくても、講座の進行はできます。

```

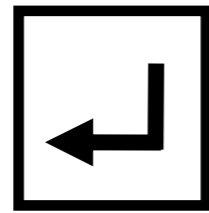
OK
RUN

HELLO
ICHIGO
JAM

OK
    
```

リスト（プログラムみせて）

LIST



F4でもOK



楽しいキャラをさがそう！

C

左下の[ALT]キー
を
おしながらC

ねこのほかにもいろいろいるよ



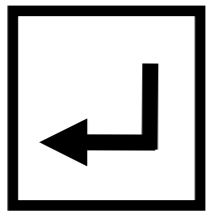
絵文字

- IchigoJamには、特殊な絵文字があります
 - 入力の方法：
 - [ALT] キーとアルファベット1文字（Cなど）を同時に押す
 - 例：
 - [ALT] + A : 10
 - [ALT] + B : オニギリ
 - [ALT] + C : ネコ
 - ...
 - [ALT] + V : イチゴ
 - WindowsやMacには、これら特殊な絵文字はありません
- 指導される方へのメッセージ
 - キャラクターの変更だけでも、子供たちの創造は広がります



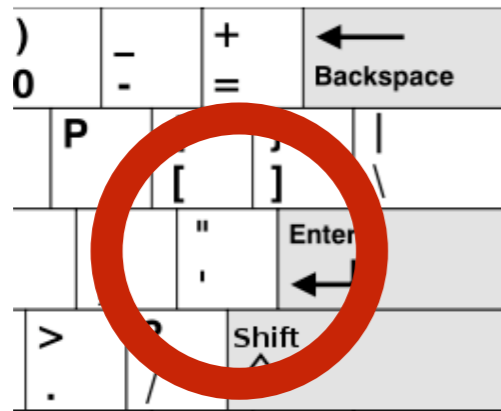
ABCDEFGHIJKLMN O PQRSTU V
10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

CLS



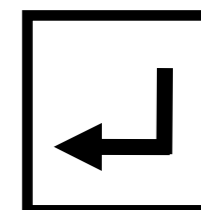
F1でもOK





「”」 (ダブルクォート) は
キーボード右のEnter左となり
SHIFTと同時に、おす

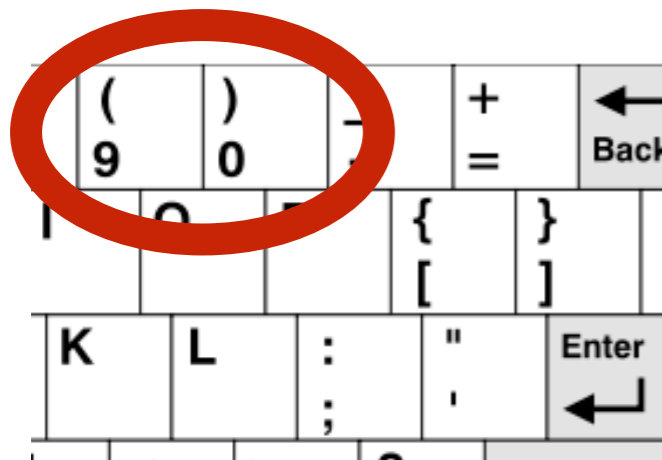
20 LC X,5:?"c"



うちこんでエンター
F5でうごかす！

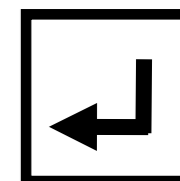
メモ：
座標 X,5 にネコが出現します。
Xは、先程入力したプログラムの通りで 16 が
入っているなので 16,5 の位置。

じぶんキャラ



「(」 「)」 (かっこ) は
SHIFTおしながら、「9」 「0」 キー
「*」 (アスタリスク) は、
SHIFTおしながら、「8」 キー

30 LC RND(32),23:?"*"



うちこんで、エンター、F5
F5をれんだすると??

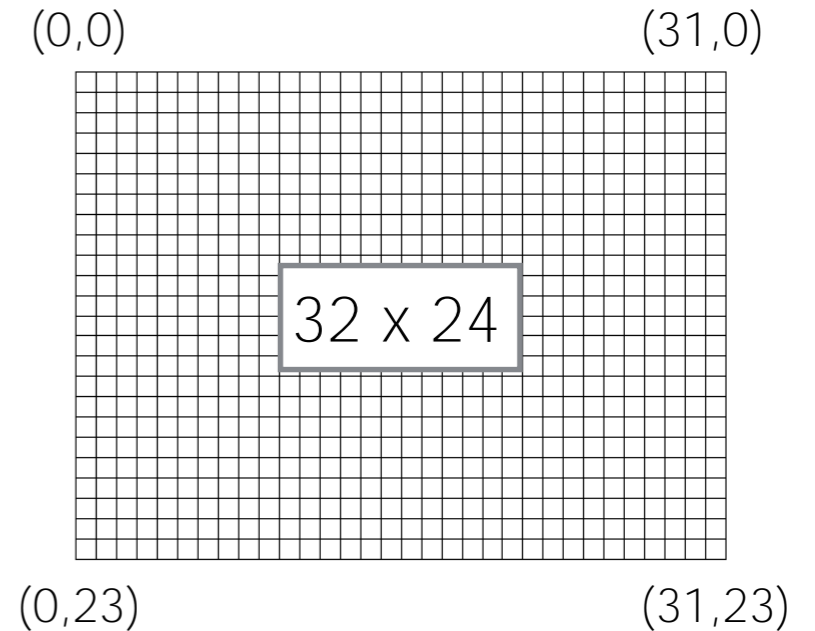
エンター・F5を連打すると、敵がランダムにあちこちに出現します。

てきキャラ

座標, LC, RND(n)

- 座標

- IchigoJamの画面は、X, Y: 2次元で位置が指定できる構造で、32×24の解像度です
- 座標系は、画面左上が原点(0,0)で、画面右下が最大地点(31,23)です。右にいくとXが増加。下にいくとYが増加します。
- 解像度の高い仕組みをとっていますが、多くのコンピュータで、同じ仕組みをとっています。



- 命令: LC X, Y

- 次のPRINTする文字を表示する場所を、座標 X, Y で指定します
- LC X, Y は、LOCATE X, Y の省略形です (どちらでも同じ動作)

- 命令: RND(n)

- nより小さな数字を、ランダムに出す命令です (乱数を取得)
- RND(10) とすると0から9までの数字を一つ、ランダムに選ぶ
- 実行するたびに引き出す数字が異なる、敵の位置を決めるなどに活用が出来ます



講座2c：ゲーム作成 -完成-

- ここまでに作った内容をベースに、ゲームを完成させます
- ゲームオーバーを作るまでは、無限に動作を行うのでESCでの止め方の指導をします
- 早く進んだ子供は、他の子をサポートしたり、先の講座を取り組ませても良いです

40 GOTO20

うちこんで、エンター、F5

てきキャラ

35 WAIT 3

とめる (ESC)

みる (F4)

うごかす (F5)

スピードちょう
せい

LIST, GOTO n, WAIT n

- 命令 : LIST

- LIST は、現在記憶しているプログラムを表示する命令です。
- ここまでのプログラム一覧は、図のよう

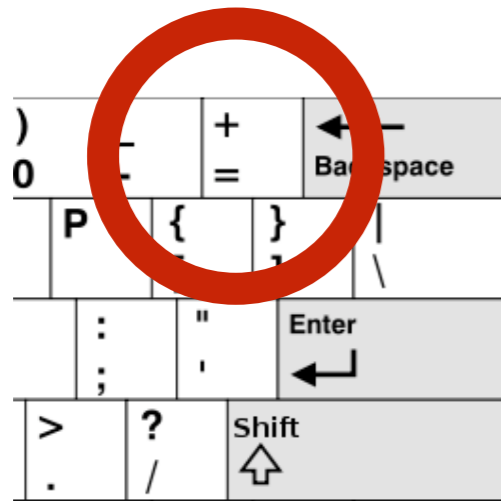
```
LIST
100 CLS : X = 16
200 LCD X : GOTO 30
300 LCD " (32) , 23 : ? " * "
400 WAIT 30
500 GOTO 20
OK
```

- 命令 : GOTO

- GOTO は、指定した数字の行へジャンプする命令です。
- この命令を使って、プログラミングの大切な要素「ループ」を作ることができます。
- ここまでのプログラムでは、10, 20, 30, 40 と実行するとまた 20 に戻り 20, 30, 40 と繰り返すという終わらないループ「無限ループ」を作っています

- 命令 : WAIT n

- WAIT は、指定した数だけ次の処理へ移るのを待つ命令です。
- WAIT 60 で 約1秒間待ちます。ですので、WAIT 300 だと 5秒待ちます



「+」 (プラス) は
キーボード右上のBS左となり
SHIFTと同時に、「=」キーをおす

36 X=X-BTN(28)+BTN(29)

とめる (ESC)
みる (F4)
うごかす (F5)

メモ：
36 LET X, X-BTN(28)+BTN(29)
BTN(28) → BTN(LEFT)
BTN(29) → BTN(RIGHT)
書き換えることも出来ます

左右キーでそう
さ

37 IF SCR(X,5) END

とめる (ESC)

みる (F4)

うごかす (F5)

あたりはんてい

矢印キー, IF, SCR(x,y)

- 矢印キー



- 次の行で、キーボードの左右矢印を使って動くようにします

- $36 X=X-BTN(28)+BTN(29)$

- 左矢印キーを押す : BTN(28) が1, 右矢印キー : BTN(29) が1
- 動作) 左が押されたら、Xを1つ減らし、Xを1つ増やす

- $36 LET X, X-BTN(28)+BTN(29)$ と書くことも出来ます。

- 命令 : IF

- IF は、指定する条件の一致・不一致により、処理を変える命令です

- この命令を使って、プログラミングの大切な要素「条件判断」を行います。

- 例) IF A==0 END の場合 変数Aに0が入っていたら、プログラムを終了します

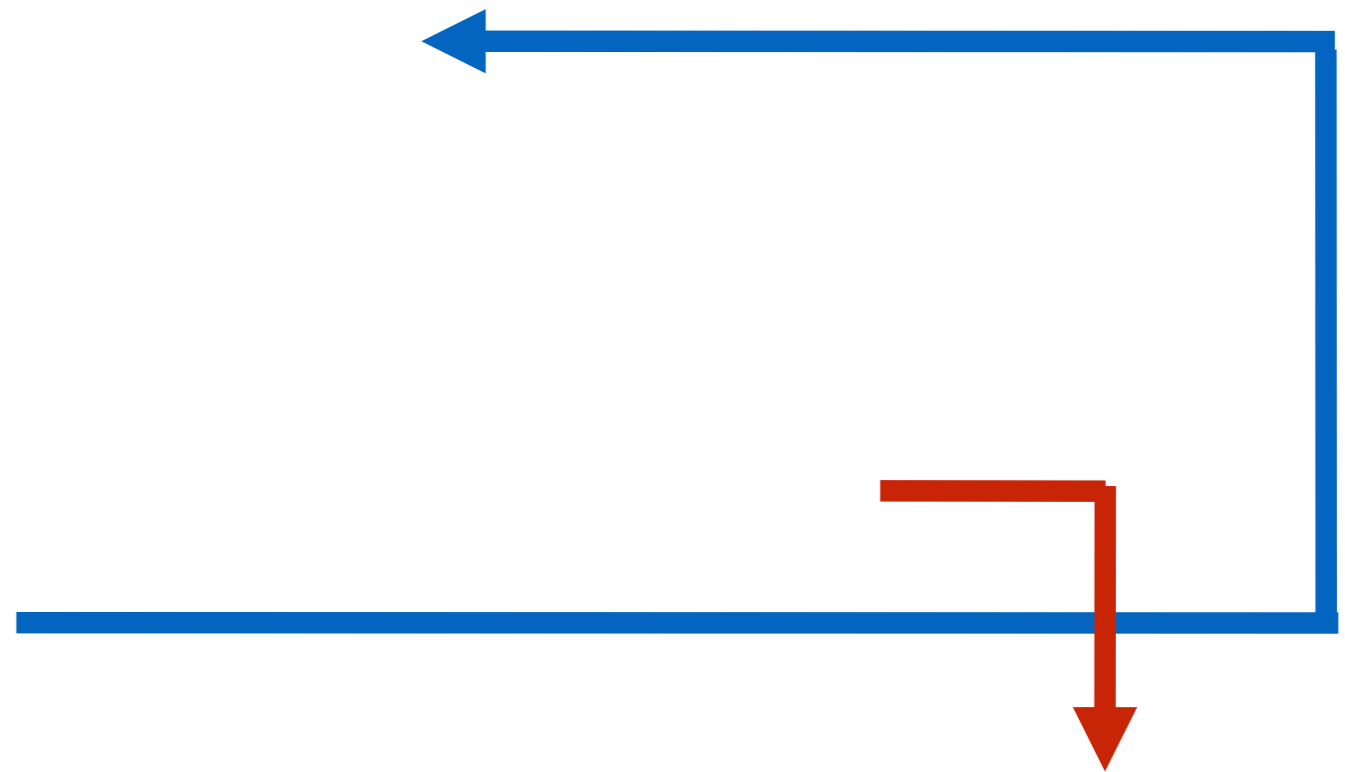
- 命令 : SCR(x,y)

- SCR は、指定した(x,y)座標にある文字の文字コードを取得する命令です。

- $37 IF SCR(X,5) END$ では、自分の位置直下(X,5)に何か文字がある(敵がいる)場合には SCR(X,5) が0ではなくなり、プログラムを終了します

- $37 IF SCR(X,5)>0 END$ や $37 IF SCR(X,5) \neq 0 END$ と書くことも出来ます。

```
10 CLS:X=16
20 LC X,5:?"c"
30 LC RND(32),23:?"*""
35 WAIT 3
36 X=X-BTN(28)+BTN(29)
37 IF SCR(X,5) END
40 GOTO 20
```



じぶんキャラのいちに
てきキャラがいたらEND

できあがり！

ほぞん（プログラムかきこみ）

SAVE1

F3 と 1でもOK





講座2d：ゲーム作成 -改造-

- 完成したゲームを、子供がオリジナルに改造します
- 改造を行いながら、プログラムで指示した理解を深めます
- キャラクターの変更だけでも、全員ができるの良いと思います
- 早くできた子供は、BEEPを使って音を鳴らすなどの取組も有効です

楽しいかいぞう



```
10 CLS:X=16
20 LC X,5:?"j"
30 LC RND(32),23:?"eee"
35 WAIT 10
36 X=X-BTN(28)+BTN(29)
37 IF SCR(X,5) END
40 GOTO 20
```

とめる (ESC)
けす (F1)
みる (F4)
うごかす (F5)

キャラをかえ
る

スピードをか
える

```
10 CLS:X=16
20 LC X,5:?"c"
30 LC RND(16),23:?"*""
35 WAIT 3
36 X=X-BTN(28)+BTN(29)
37 IF SCR(X,5) END
40 GOTO 20
```

とめる (ESC)
けす (F1)
みる (F4)
うごかす (F5)

テキが半分に
?

5 CLT

```
10 CLS:X=16
```

```
20 LC X,5:?"c"
```

```
30 LC RND(32),23:?"*"
```

```
35 WAIT 3
```

```
36 X=X-BTN(28)+BTN(29)
```

```
37 IF SCR(X,5) ?TICK(): END
```

```
40 GOTO 20
```

メンター・
コアが表示される
でる数字と表示する
)にする

・終了時に、何カウント逃げれたかのスコアが表示される
・TICK()は、JchigoJam内部でカウントされる数字と表示する
・CLTは、TICK()のカウントを初期化（ゼロ）にする

スコアを出す

あれ、みっかかった？



```

10 CLS:X=16
20 LC X,5:"?"j"
30 LC RND(32),23:"?"eee"
35 WAIT 3
36 X=X-BTN(28)+BTN(29)
37 IF SCR(X,5) END
38 X=(X+32)%32
40 GOTO 20

```

メンター
 ・画面左にぶつかると、右に出てくる
 ・画面右にぶつかると、左に出てくる
 ・%は剰余演算（割り算の余り）

剰余演算の例)

$32\%32 = 0$ ($32\div32 = 1$ あまり 0)
 $33\%32 = 1$ ($33\div32 = 1$ あまり 1)
 $34\%32 = 2$ ($34\div32 = 1$ あまり 2)
 $62\%32 = 30$ ($62\div32 = 1$ あまり 30)
 $63\%32 = 31$ ($63\div32 = 1$ あまり 31)

$(5\div4 = 1$ あまり 1)
 $\cdot 3\%32 = 31$ ($5\div4 = 1$ あまり 1) $\cdot 5\%2 = 1$
 $(5\div1 = 1$ あまり 1)

チートたいさく
1

```
10 CLS:X=16
20 LC X,5:?"j"
30 LC RND(32),23:?"eee"
35 WAIT 3
36 X=X-BTN(28)+BTN(29)
37 IF SCR(X,5) END
38 IF X<0 OR X>31 END
40 GOTO 20
```

メンター
画面
左右にぶつかったら、ゲームオーバーになる

チートたいさく
2

ほぞん

(つくったプログラムかきこみ)

SAVE1

F3、1でも
OK



よみこみ

(ほぞんしたプログラムをよみこむ)

LOAD1

F2、1でも
OK



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押しながらで切り替え）、'[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 / キャップス	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:FREE(), F7:OUT0、F8:VIDEO1、F9:FILES
ボタン	押しながら起動でFILE0を自動実行する

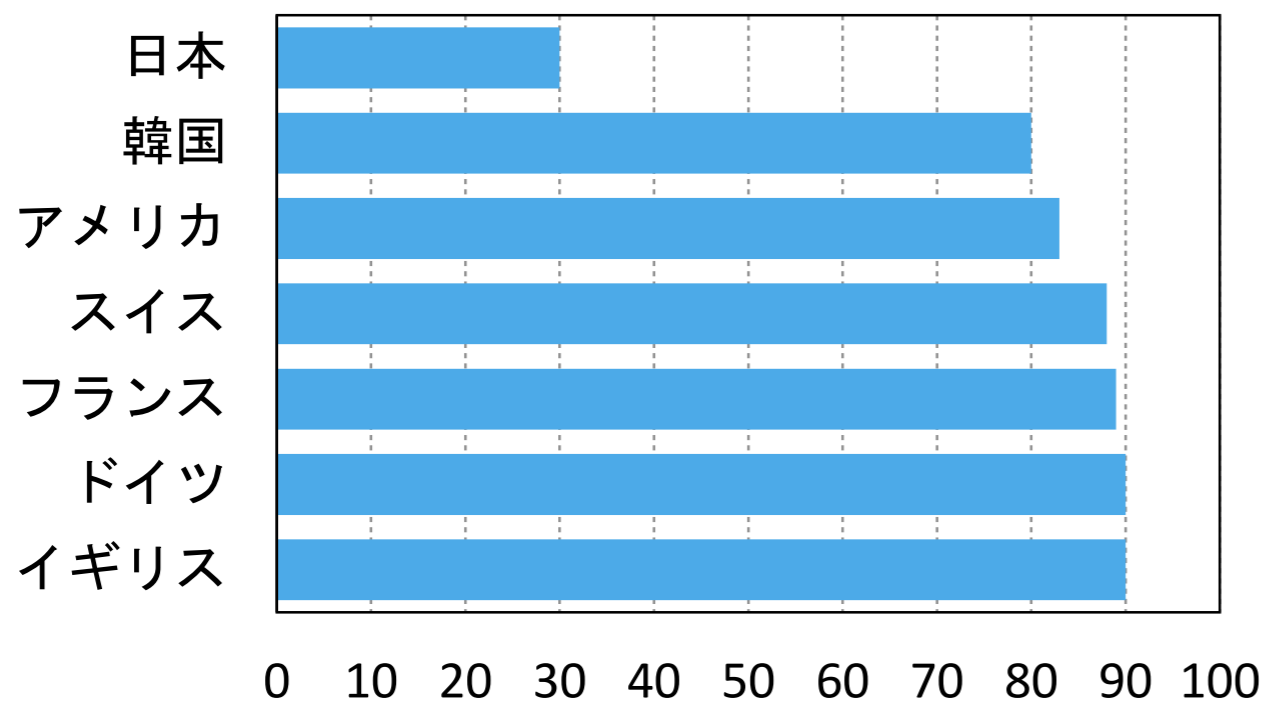
初級コマンド

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で待つ(-261でWAIT1と同等)	WAIT 60
: / コロン	コマンドを連結する	WAIT 60:LED 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の時終わりまで表示、ESCで途中停止）	LIST 10,300
GOTO 行番号 / ゴートゥー	指定した行番号へ飛ぶ（式も指定可能）	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 BTN() END
BTN((数)) / ボタン	ボタンが押されていれば1、そうで無いたま0を返す（数：0(付属ボタン)/UP/DOWN/RIGHT/LEFT/SPACE、省略で0）	LED BTN()
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 (小数点以下は切り捨て)	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文字を変数として数の値を入れる (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からの入力で数値を変数に入れる (文字列とコンマは省略可)	INPUT "ANS?",A
TICK() / ティック	CLTからの時間を返す (約1/60秒で1進む)	PRINT TICK()
CLT / クリアティック	時間をリセットする	CLT
INKEY() / インキー	キーボードやUARTから1文字入力する (入力がない時は0、UARTから0が入力された時は#100)	PRINT INKEY()
CHR\$(数) / キャラ	文字コードに対応する文字を返す (コンマ区切りで連続表記可)	PRINT CHR\$(65)
ASC("文字") / アスキー	文字に対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
SCR((数,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をいれ、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:?!: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を省略で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 ーエム	外部出力OUT2-5に数2で0.01msec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、マイナス値指定で周期1/480)	PWM 2,100

作る と 遊ぶ

- 「ゲームを作る」と「ゲームを遊ぶ」の間には、大きな差があります。数学（動作を作る為）をプログラミングで組み合わせる能力が働きます。
- 日本は、若年層のパソコン保有率が極めて低い（図）。かわりに、ゲーム機、タブレットなどの所有率が高くなっています。
- コンピューターを利用したモノづくり・ソフトウェア作りを行うには、まだがまきです。将来的な、日本国の課題解決力の不足に繋がる懸念があります。



13～15歳 自分のパソコン保有率

内閣府「わが国と諸外国の若者の意識に関する調査」(2013年)より



プログラミング クラブ ネットワーク

すべてのこどもたちに
プログラミングを

<http://pcn.club/>