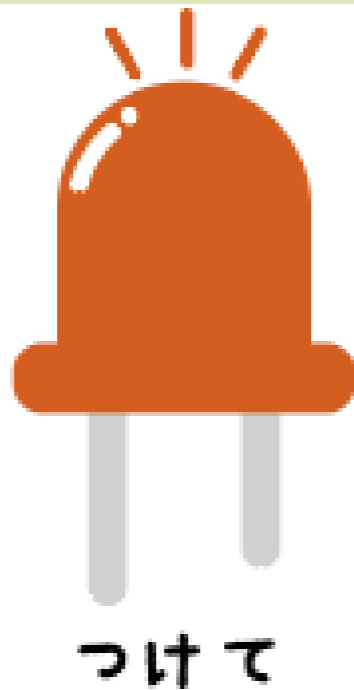


# はじめのいっぽ



# 導入、プログラミング体験 プログラミングをなぜ学ぶ？

この講座では、次の事柄を、IchigoJamを実際に操作して体感しながら、学んでいきます。

- プログラミングとは？
- 人間もコンピューターも、互いの言葉を知らない
- コンピューターの得意な事：計算、繰返し、記憶
- プログラミング的思考を学ぶ
- 生活の中の、コンピューターの応用分野を発見する

# 導入、プログラミング体験

## プログラミングをなぜ学ぶ？

- プログラミングとは？  
= コンピューター (IchigoJam) に仕事をお願いする事
- 人間もコンピューターも、互いの言葉を知らない  
→ カードを使ったり、資料をみてOK  
→ IchigoJamはBASIC言語を使う。英単語に近い言葉
- コンピューターの得意な事：計算、繰返し、記憶  
→ 計算のスピードは？  
→ 何回繰返しさせる？  
→ 記憶したものは、いつまで覚えている？
- プログラミング的思考を学ぶ  
→ 大切なのは、仕事の伝え方の手順とそれを考えること  
→ 命令などの暗記ではなく、組み立て方を重視  
→ 言語 (IchigoJamの場合：BASIC) は、やりたい事に応じて変化
- 生活の中の、コンピューターの応用分野を発見する  
→ プログラムはプログラマーだけのものではなく、学んだ人が適材適所で使える技術。今後も、あらゆるところに応用できる

# 導入、プログラミング体験 プログラミングをなぜ学ぶ？

- スケジュール
  - 講座1a: 導入 -接続と操作-
  - 講座1b: 順次処理とスピード
  - 講座1c: 確認問題と発展問題
  - 講座1d: 生活と学び方、まとめ
- 機材一覧
  - IchigoJam, キーボード, 電源, テレビ, 各種ケーブル
  - カトラリーカード, 台紙
- 指導される方へのメッセージ
  - 答えは複数あるので、書き方が異なっていてもOK
  - 間違いから学ぶので、間違ってもOK
  - 教え合えるものなので、教えあってもOK



# 講座1a：導入 -接続と操作-

- 目の前にあるIchigoJamを、まずは触ってみます
- まだ動作させるすべを知らないなので、カードを使って翻訳
- 触る流れの中で、人とコンピューターの言葉の違いなどを学びます

プログラミング

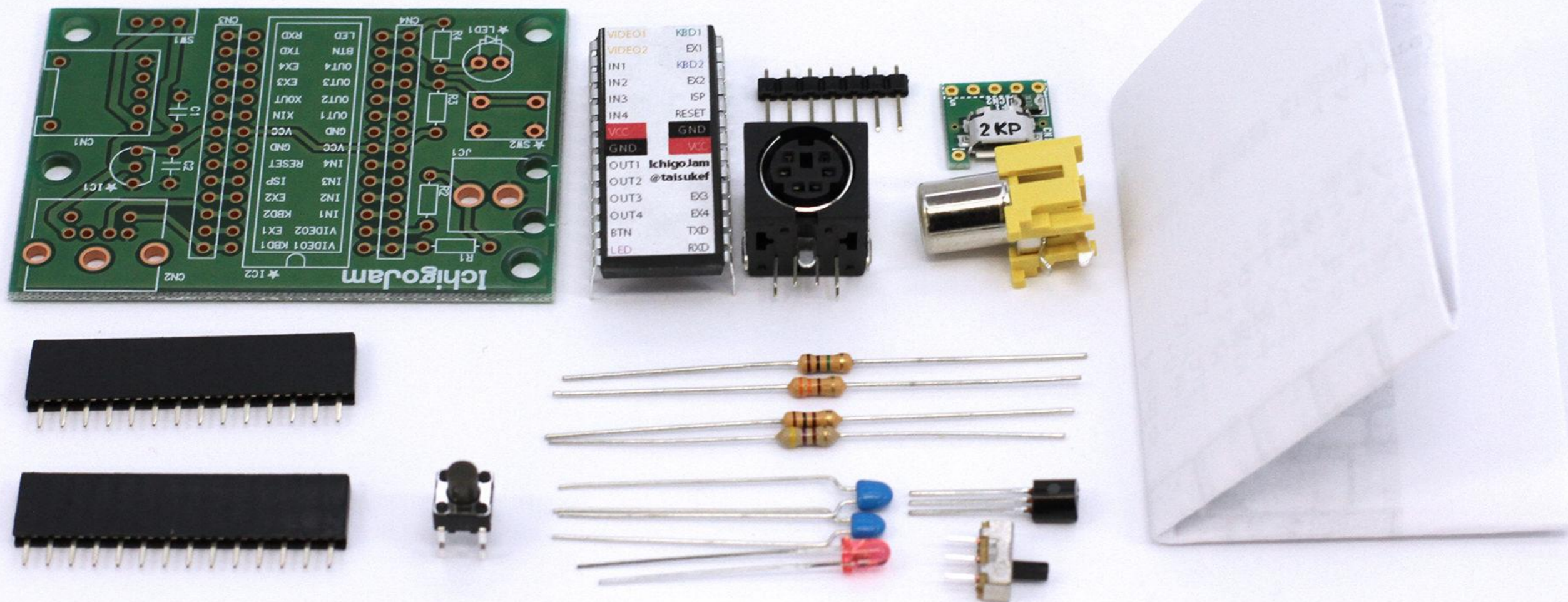
=

コンピューターにおしえること





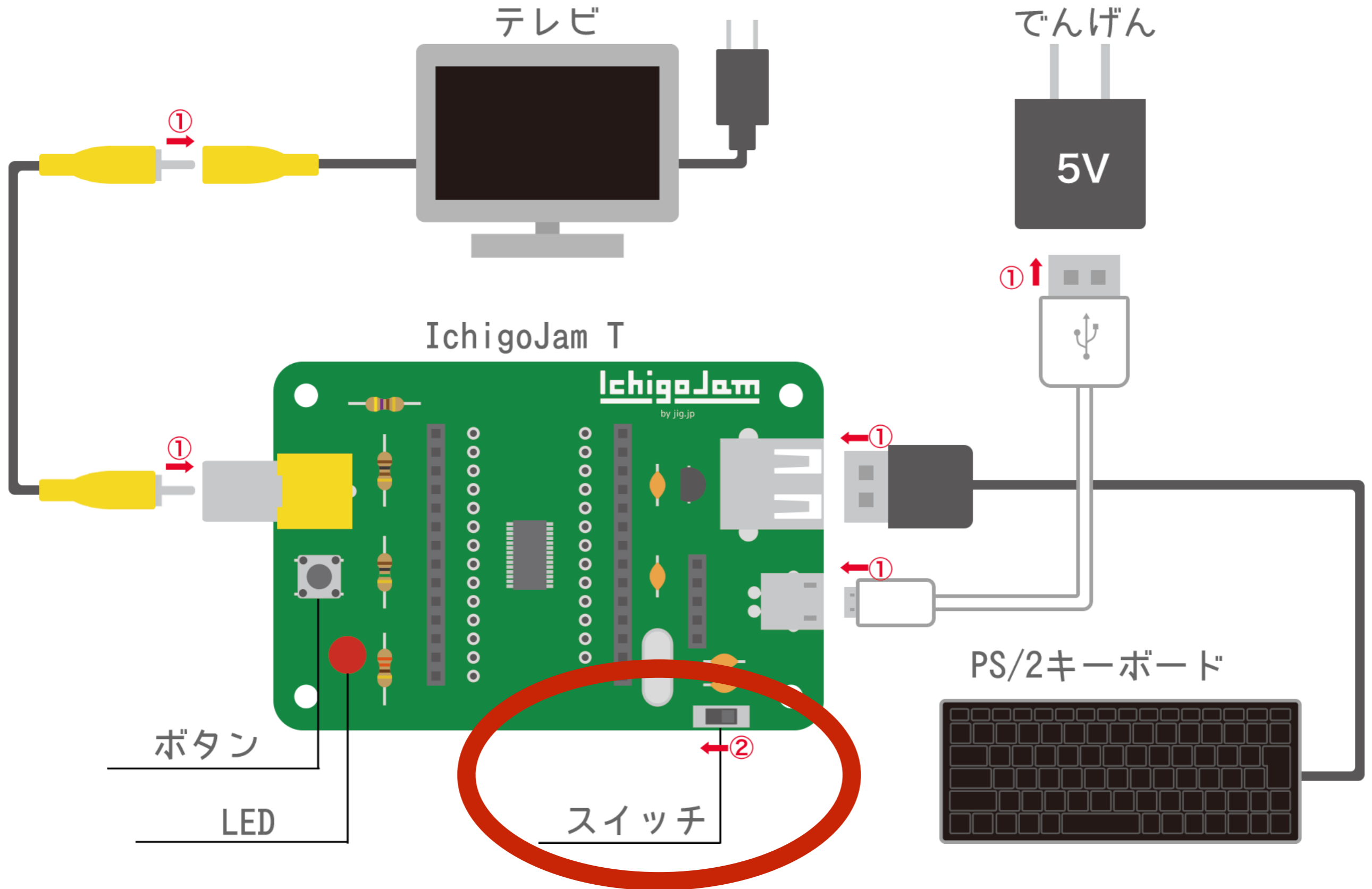
# IchigoJam



じぶんでくみたてるパソコン

IchigoJam 1,500円

# つないで、でんげんON





# IchigoJam BASIC

OK

|

メモ：

- IchigoJamは、英語「大文字」  
入力で、キーボード上の文字  
そのまま。  
Windowsなどは「小文字」

てんめつしているのは、カーソル

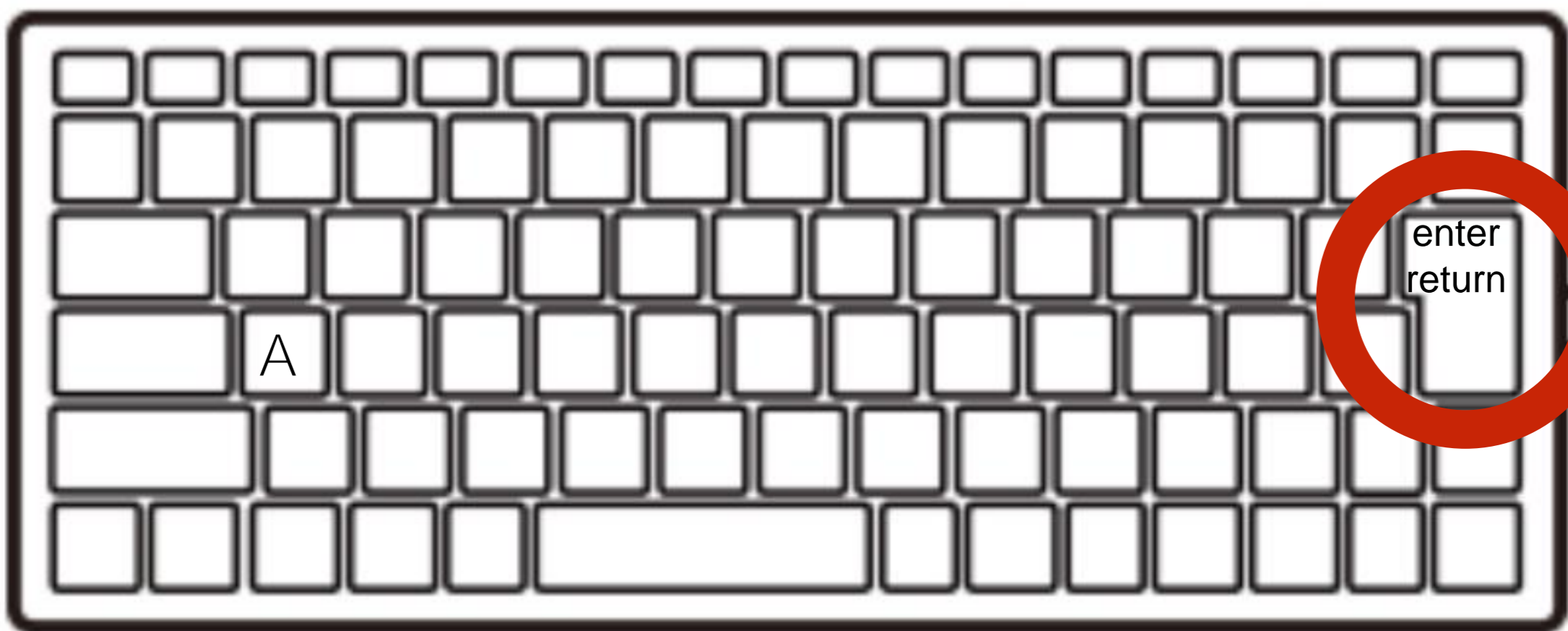
キーボードの“A”をうってみよう

# IchigoJam BASIC

OK

A

Syntax error



Aの後は、 エンターキー

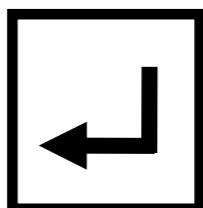
メンター：

- コンピューターに、キーボードを通じて言葉を伝える



シラナイ  
コトバダ

A



(エー、エンター)

Syntax error

(シンタックスエラー)

# ほんやくカード！

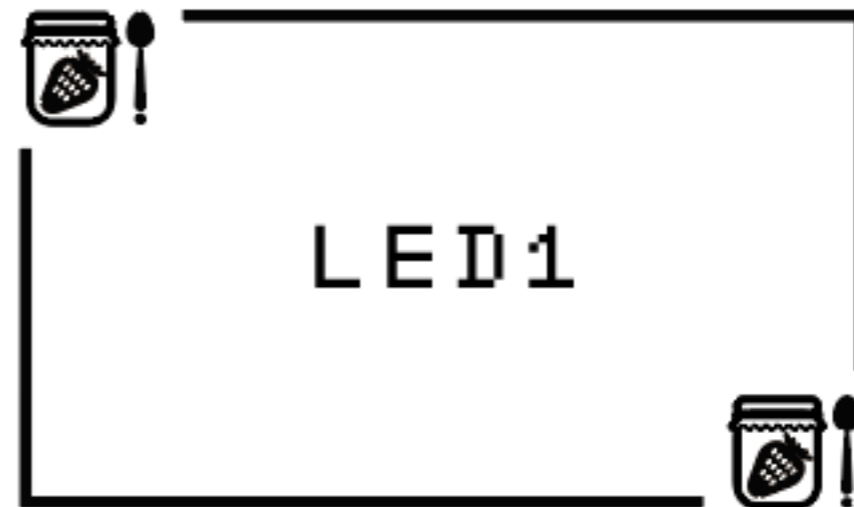


# LEDをつけよう



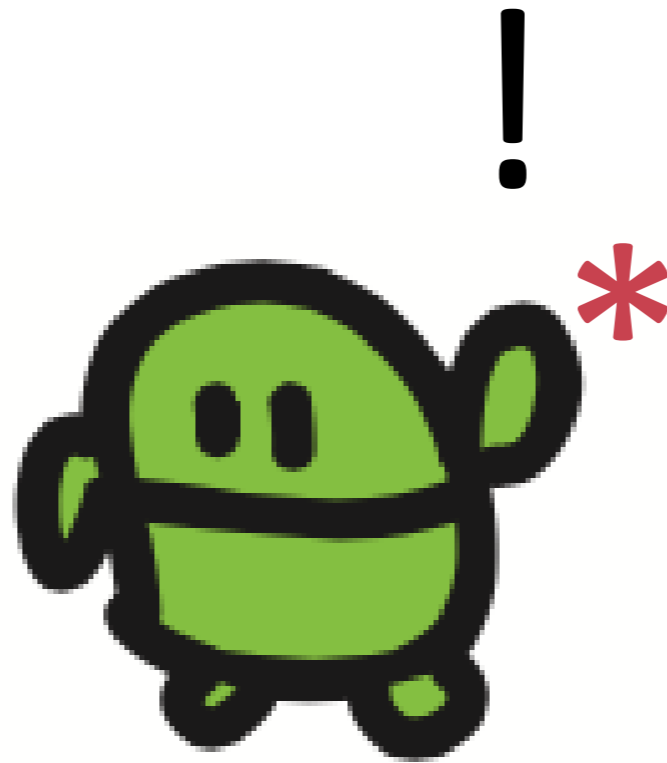
メンター：

- カードを探して、裏返す



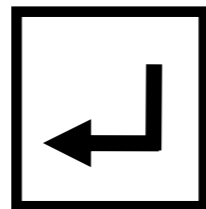
ひっくりかえすと、  
IchigoJamのコトバに！





シッテル！

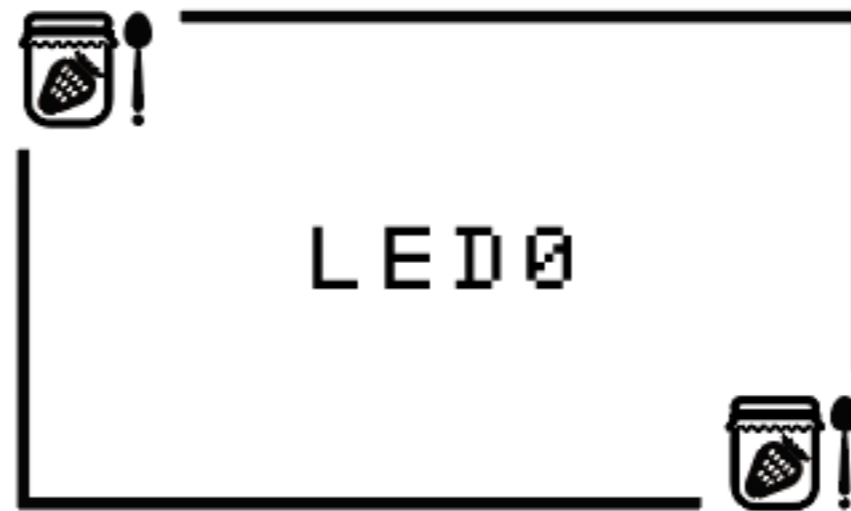
LED 1



(エリイーディーいち、エンター)

OK  
(オーケー)

# LEDをけそう

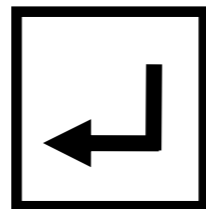




メンター：

- LED だけを入力してみたり、どんな「言葉」が理解されるか試す
- 遠慮なく間違っても良いとのメッセージを子供たちに

LED 0



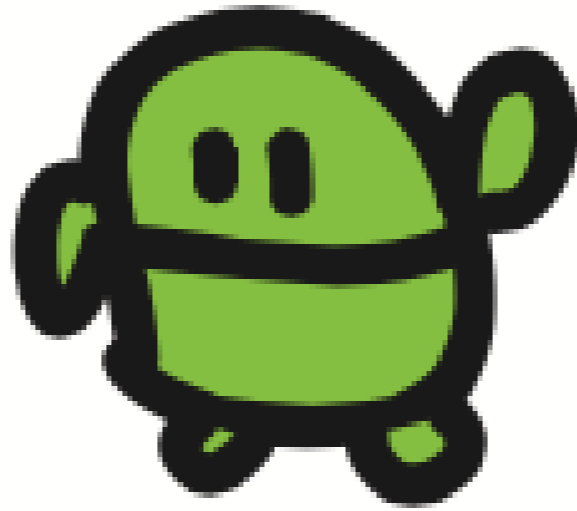
(エリイーディーいち、エンター)

OK



# 講座1b：順次処理とスピード

- プログラムが書かれた順番で動作していく、順次処理を学びます
- また、コンピューターの動作速度を体感して、計算スピードを体感することができます。



メンター：

- 複数のステップを、コンピューターに、まとめてやらせてもらおう。

LEDを  
つけて、けそう



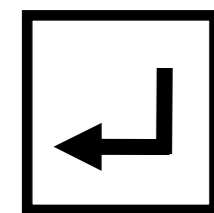
10



つけて



LED1



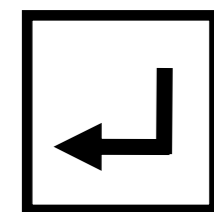
20



けして



LED0



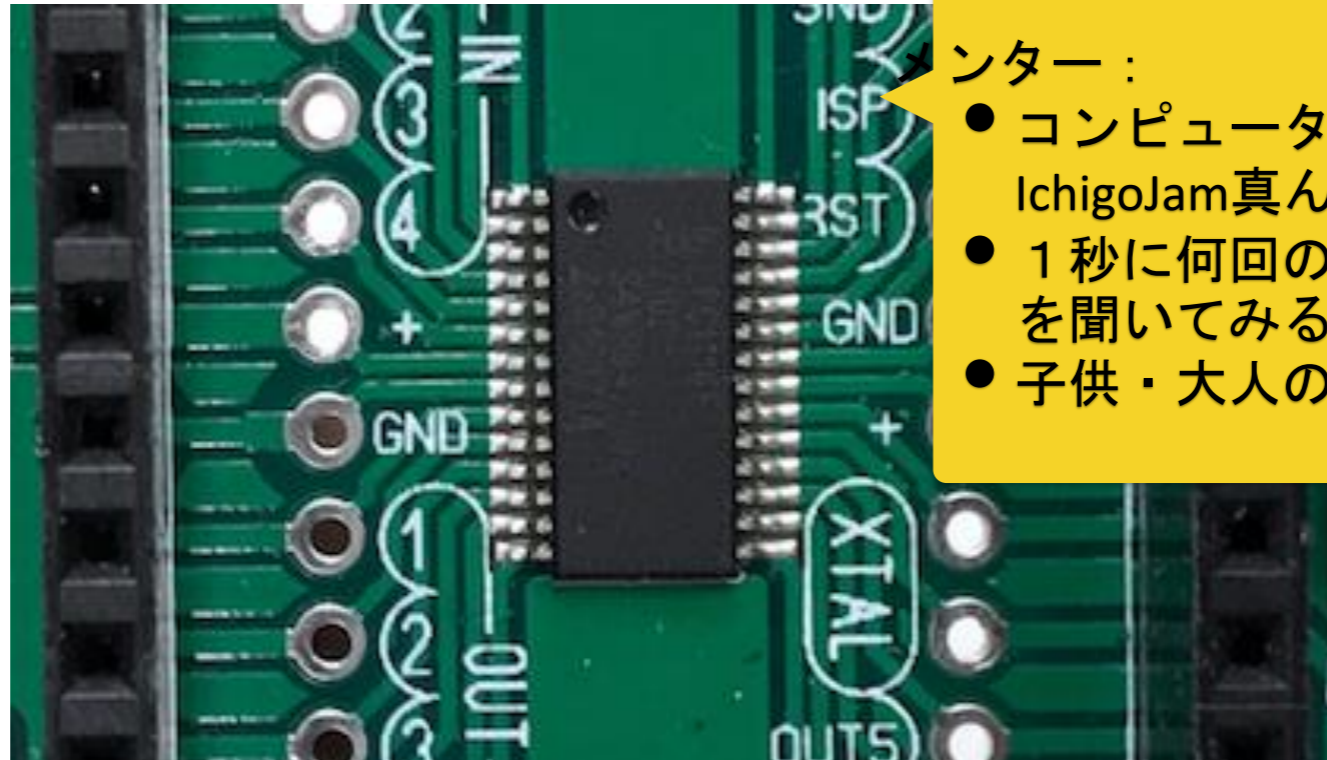
メンター：

- 台紙にカードをならべて、台紙上の数字10から続けて入力します
- 最後にF5キーをおすと、入力したプログラムが一気に実行

おや？



F5キー



インター :

- コンピューターの頭脳CPUを探す (IchigoJam真ん中にあるIC)
- 1秒に何回の足し算ができるだろうかを聞いてみる (答えは後述の表)
- 子供・大人の想定を大きく超える

150円のコンピューター  
1秒に何回、足し算できる？

1秒に「回」

ちょうどいい



10



つけて

20



1びょうまって

30



けして

メンター：

- 高速に処理をおこなうため、待ちをいれて調整をする
- カードを裏返して入力したら、F5キーで実行

メンター：

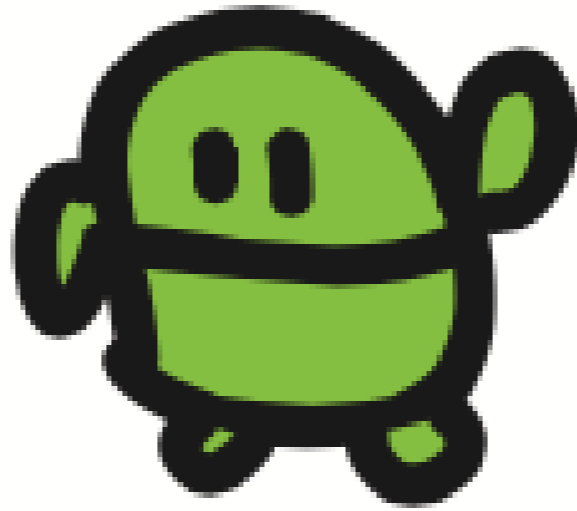
- コンピューターは記憶力も抜群。LIST (F4) を使えば、いつでも覚えてるものを表示
- WAIT 60 = 1秒なので、3秒 (= WAIT 180) など、数字を変えて確認することも大切

F5キー



# 講座1c：確認問題と発展問題

- これまでに行った操作の確認を含むクイズに取り組みます
- また、新しい操作を導入して、発展的な問題に取り組みます
- 発展問題により、繰返しの考え方を学びます



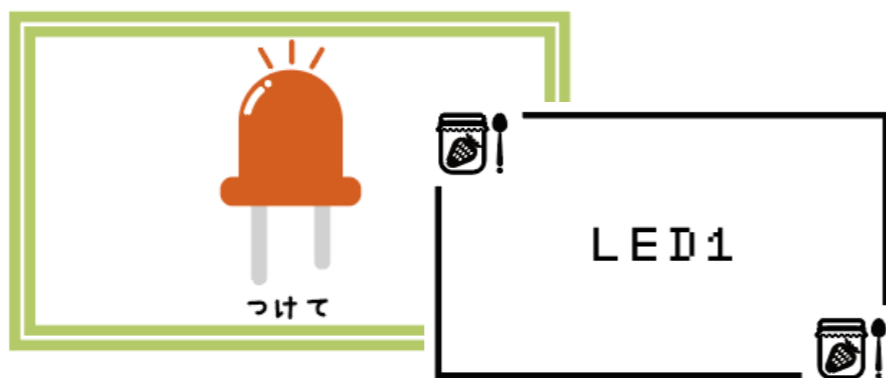
メンター：

- 講座1b で学んだ事を、応用してみる。使うカードは同じ。
- WAIT 60 = 1秒なので、数字を変えて確認することも有効

クイズ1：  
LEDを2かい  
チカチカ 光らせてみよう



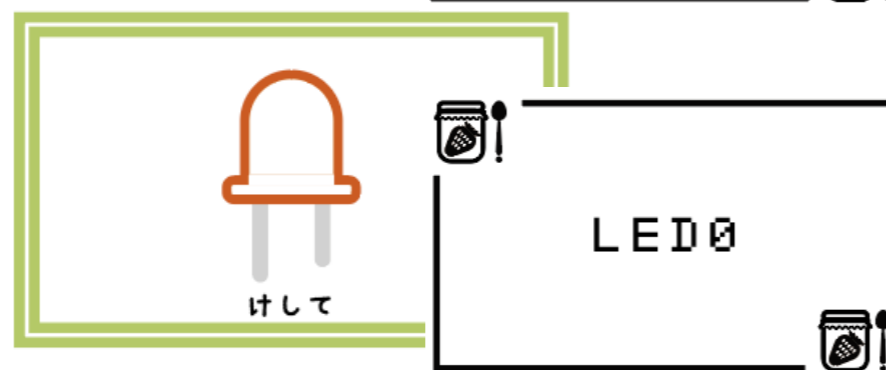
10



20



30



40



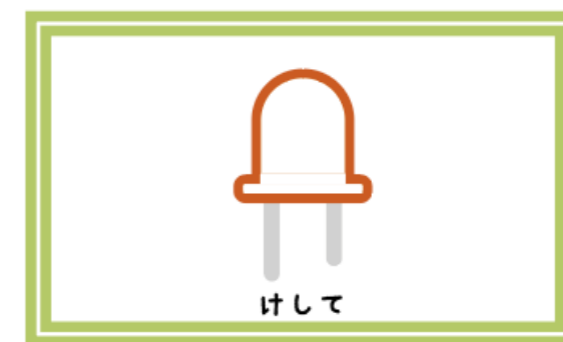
50



60



70



メンター：

- WAIT60のカード不足は適当に
- 40のWAIT60を忘れ、一回しか光らないように見える（大事）
- LED2のような入力間違いで消えない
- enterの押し忘れないか？



メンター：

- 新しいカードも使い、よりコンピューターらしい仕事をさせる
- 今まで延長で100回点減させる場合に必要なプログラムと比較

クイズ2：  
LEDをずっと  
てんめつさせしてみよう

# 新しく使うカード

メンター：

- コンピューターの得意な「繰り返し」。一度決めた処理を、何度でも指定しただけ繰り返してくれます。



## くりかえし

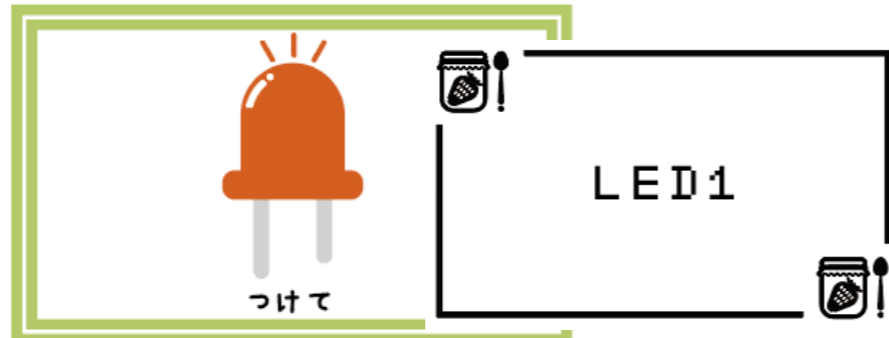


5



メンター：  
 ●無限ループ(ずっと終わらない処理)なので、ESCキーを押して、プログラムを停止させる

10



30



20



40



50



メンター：  
 ●5の場所はないが、あると想定 (打ち直し不要で進行スムーズ)  
 ●前のプログラム: 50, 60, 70は、空打ちでもOK (消さずに上書きでもOK)

クイズ2の答え  
 一例

## 編集操作 と 自動起動（+1 STEP）

- プログラム編集操作
  - LIST コマンド (F4) でプログラムを表示
  - 矢印キーにより、修正したい箇所へ移動して修正が可能
  - 修正後には、enter/return キーを押して変更をIchigoJamに教える
- ファンクションキーによる、ショートカット（キーボード一番上）
  - F1 (CLS) ... 画面上の文字が全部消す（保存内容に影響なし）
  - F3 + 0 (= SAVE 0) ... 0番目に保存（0から3まで4つ保存可能）
  - F2 + 0 (= LOAD 0) ... 0番目を読み込
- 自動起動
  - <方法>
    - SAVE0して 電源OFF
    - IchigoJam本体のボタンを押しながら 電源ON
  - SAVE 0 された内容を、電源ONと共に実行する機能
  - キーボード、ディスプレイは不要（電源OFF時に外してOK）

メンター：

- ・身の周りの家電に、コンピューターが入って動作している具体的なイメージに繋がる



# クイズ3： お家のコンピューターを かぞえてみよう

# お家のコンピューターをかぞえてみよう

- 台所
  - 電子レンジ
  - 冷蔵庫（温度測定・管理など）
  - IHヒーター
- 居間
  - 液晶テレビ（チャンネル管理、色・音の管理など）
  - HDDレコーダー（録画の仕組み・管理）
- 部屋
  - Windows / Mac などのパソコン
  - タブレットやスマホ
  - CDプレイヤーやMP3プレイヤー
- その他
  - 洗濯機

メンター：

● <探すヒント> 液晶が液晶ディスプレイ、キーボードも、タッチが沢山のもの



# 講座1d：生活と学び方、まとめ

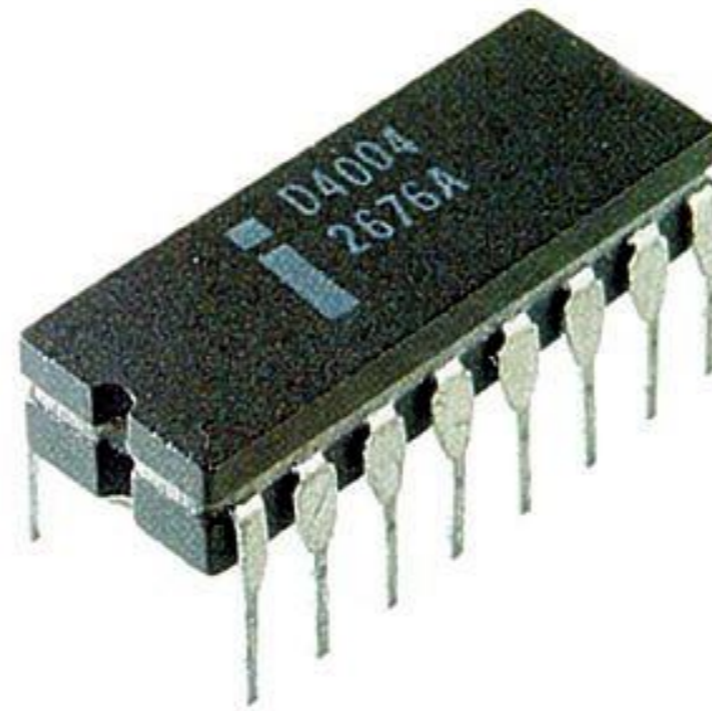
- ここまで扱ったコンピューター（IchigoJam）と、生活の接点を考えます
- IchigoJamを学ぶ方法を紹介します
- 最後に、まとめを行って講座は終了です



コンピューターと  
プログラミングと  
せいかつ



お家や学校のコンピューターそれらはみんな、人が  
プログラミングしたものの



# CPUをはつめいした日本人

CPUの元祖4004開発者は, 嶋正利さん



- メンター：  
 ● コンピューターが計算が得意な事を確認  
 <確認方法>  
 ● 小学生や大人の計算速度と比較  
 ● 触れているIchigoJamと比較  
 ● 計算ゲームを作って競争

(C)TSUKUMO



from Wikipedia

IchigoJam

iPhone

パソコン

スパコン京

5000万回

200億回

10兆回

1京回

1台分

400台分

20万台分

2億台分

1500円

7万円

10万円

1120億円



メガネ型コンピューター  
Google Glass



自動お料理ロボット  
Moley Robotics



プログラミングであやつる

プログラミングでまなべる  
”まなびかた”





# 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

<http://ichigojam.net/IchigoJam.html>

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(数)で音を出す。*SOUND(EX2)-GNDに圧電サ	BEEP
PLAY (音階)	音階を演奏する。音階はCDEFGAの順に1-9まで指定可。音階指定後、音階指定の音階を演奏する。音階指定後、音階指定の音階を演奏する。	PLAY "SCDE2CDE2"
		MP0 1200

IchigoJam  
100のコマンド



MML (PLAYコマンド内)



じぶんでまなぼう

## 学び方について

- コンピューターへのプログラミング
  - 試行錯誤が可能。間違える事は大切で、そこから学びが生まれる
  - 新しい変化が常にあるので、動かしてみる事が大事
    - 楽譜を聴く、楽器を弾く、料理を作る、プログラミングなど、興味のある分野でも挑戦的
- 学ぶための素材を確保
  - 本屋さん・図書館などで、書籍を入手する
  - オンラインの教材を探す
    - Webページ、PDF公開、YouTubeなどでの動画
- 方向性を見つける
  - 好きなもの、解決したい課題へ向かって学ぶのが良い
    - 音楽好き ≡ 曲を作る、理科好き ≡ 電子回路と組み立て
  - 1つのプログラミング言語は万能ではない
    - 基礎を学べたら、学びたい事ができる環境へとステップアップ

プログラミング

=

てじゅんをかんがえる

+

コンピューターにつたえる





プログラミング クラブ ネットワーク

すべての子どもたちに  
プログラミングを

<http://pcn.club/>