

IoTプログラミング

はじめのいっぽ

with IchigoJam × sakura.io



福井高専 電子情報工学科卒→起業



小3でプログラミングに出会う → IchigoJam開発

コンピューターは
どこにいる？



IoT、どんどんふえる
コンピューターとネット



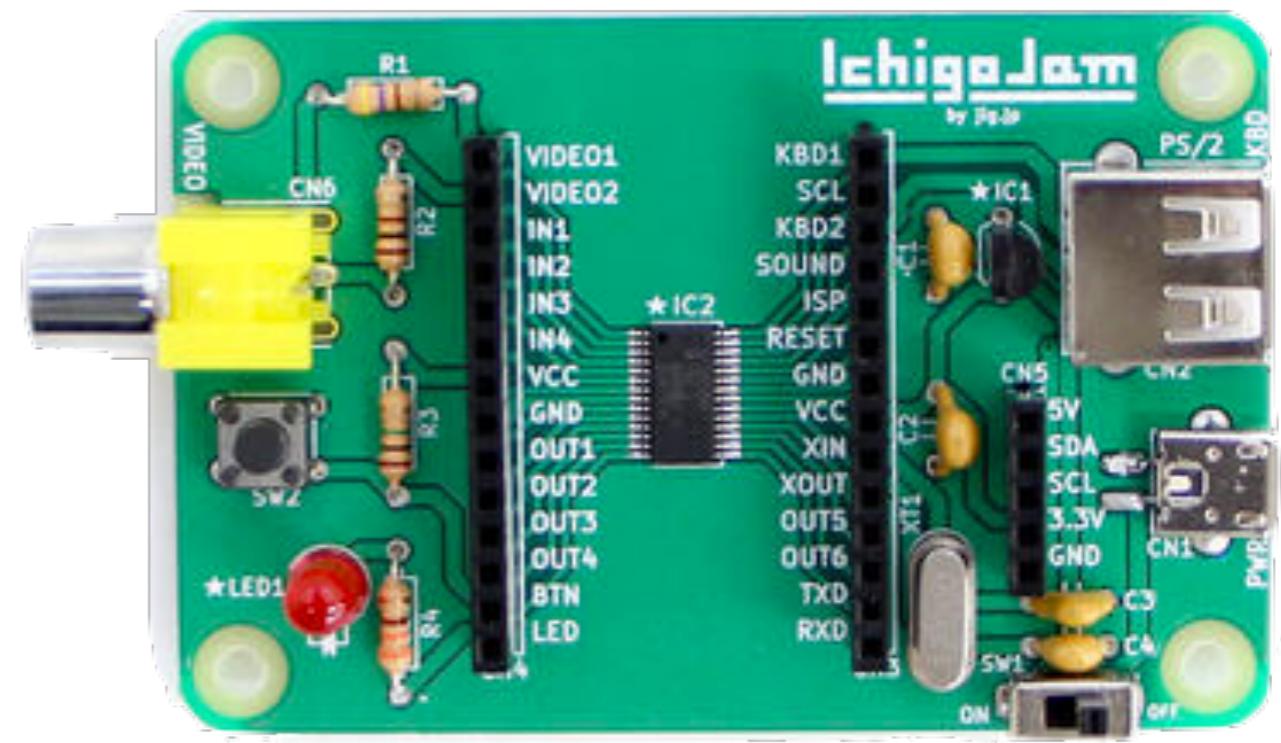
コンピューターと
なかよくなろう！



こどもパソコン IchigoJam

イチゴ"ジャム

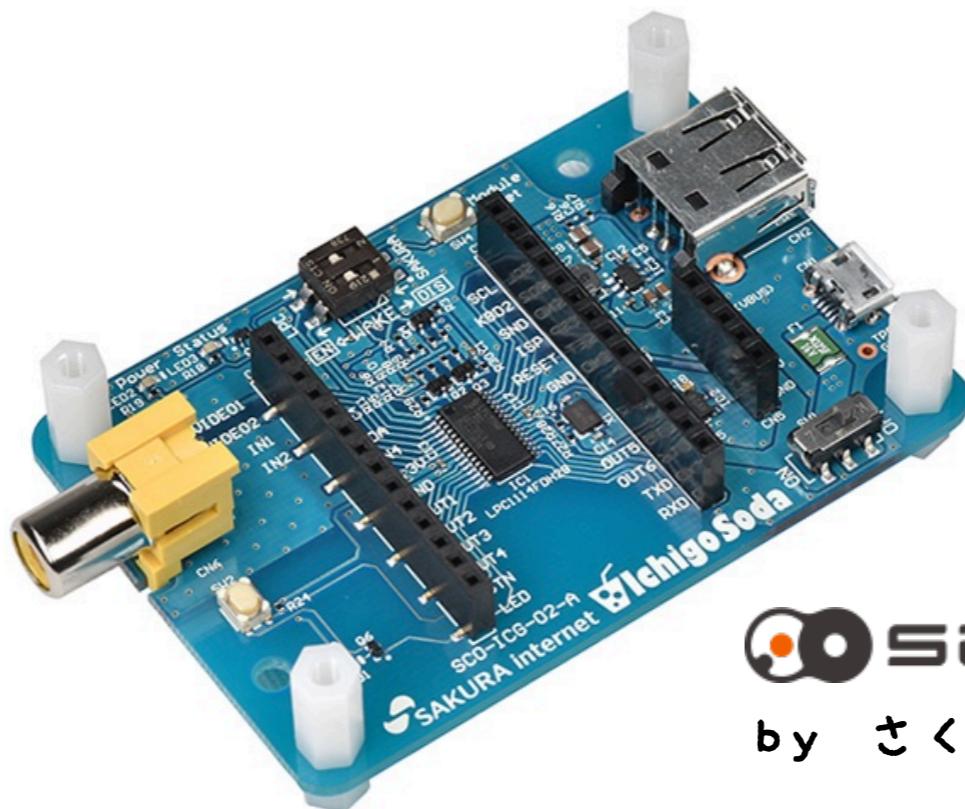
IchigoJam



おねだんイチゴー (1,500円~)

プログラミングで"ジャムセッション！

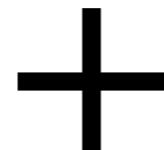
ネットにつながる IchigoJam IoT ≡ IchigoSoda



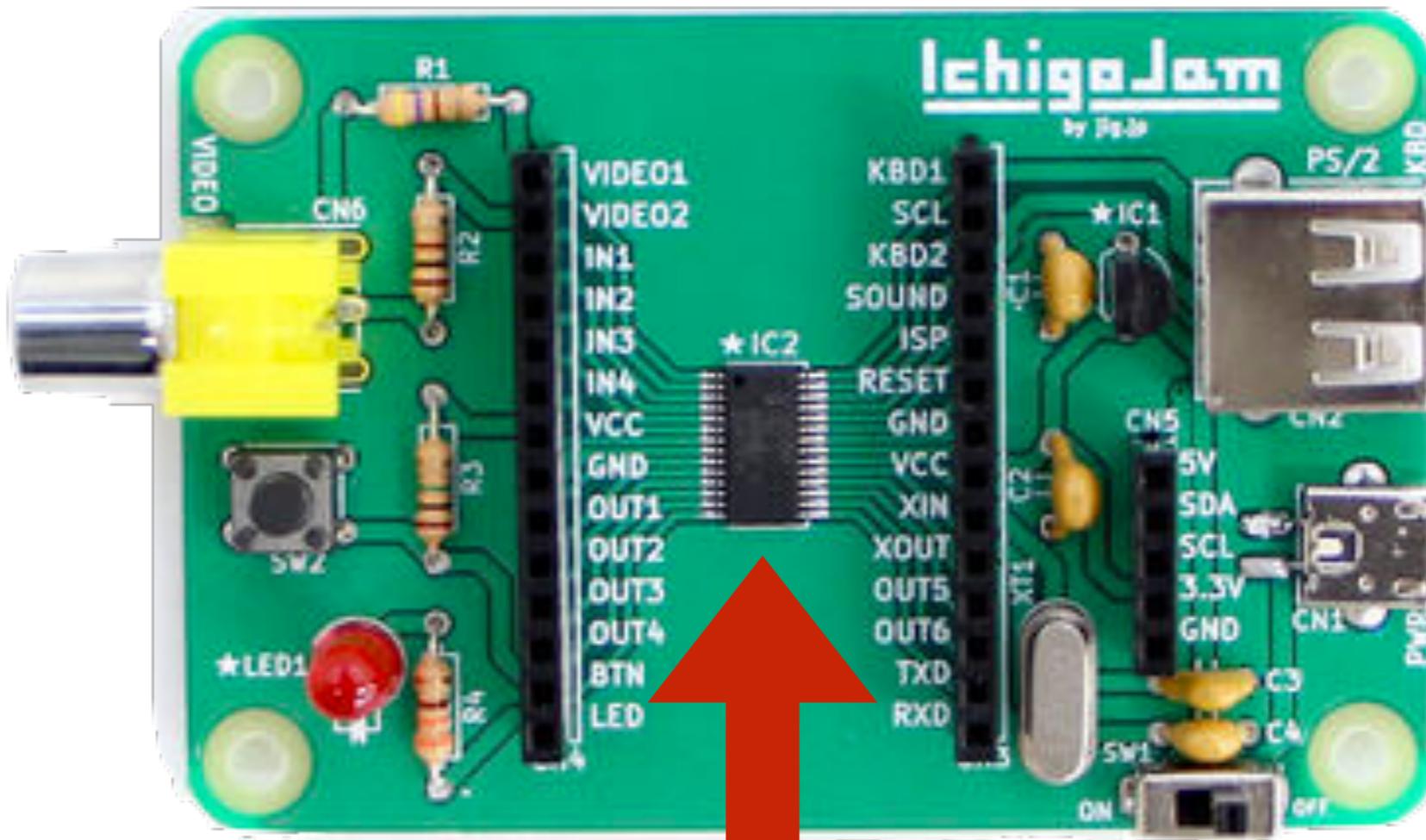
sakura.io
by さくらインターネット



IchigoJam



つうしんモジュール



これがコンピューター！

おねだん、100円！

電子工作はじめのいっぽ

LEDをつなごう

みじかいほうを
したから3ばんめ
“OUT4”



ながいほうを
いちばんした“LED”

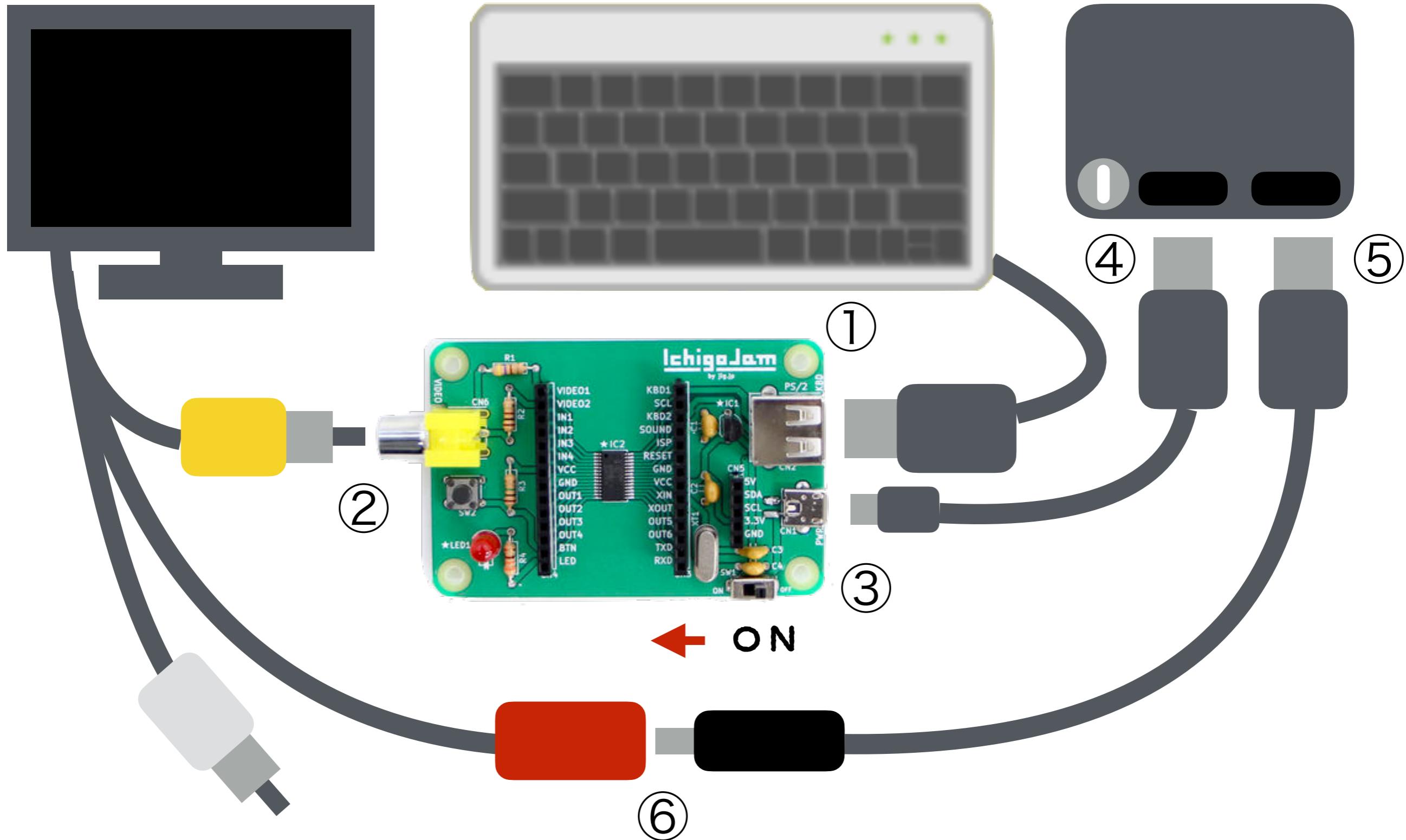


IchigoJam をつなごう

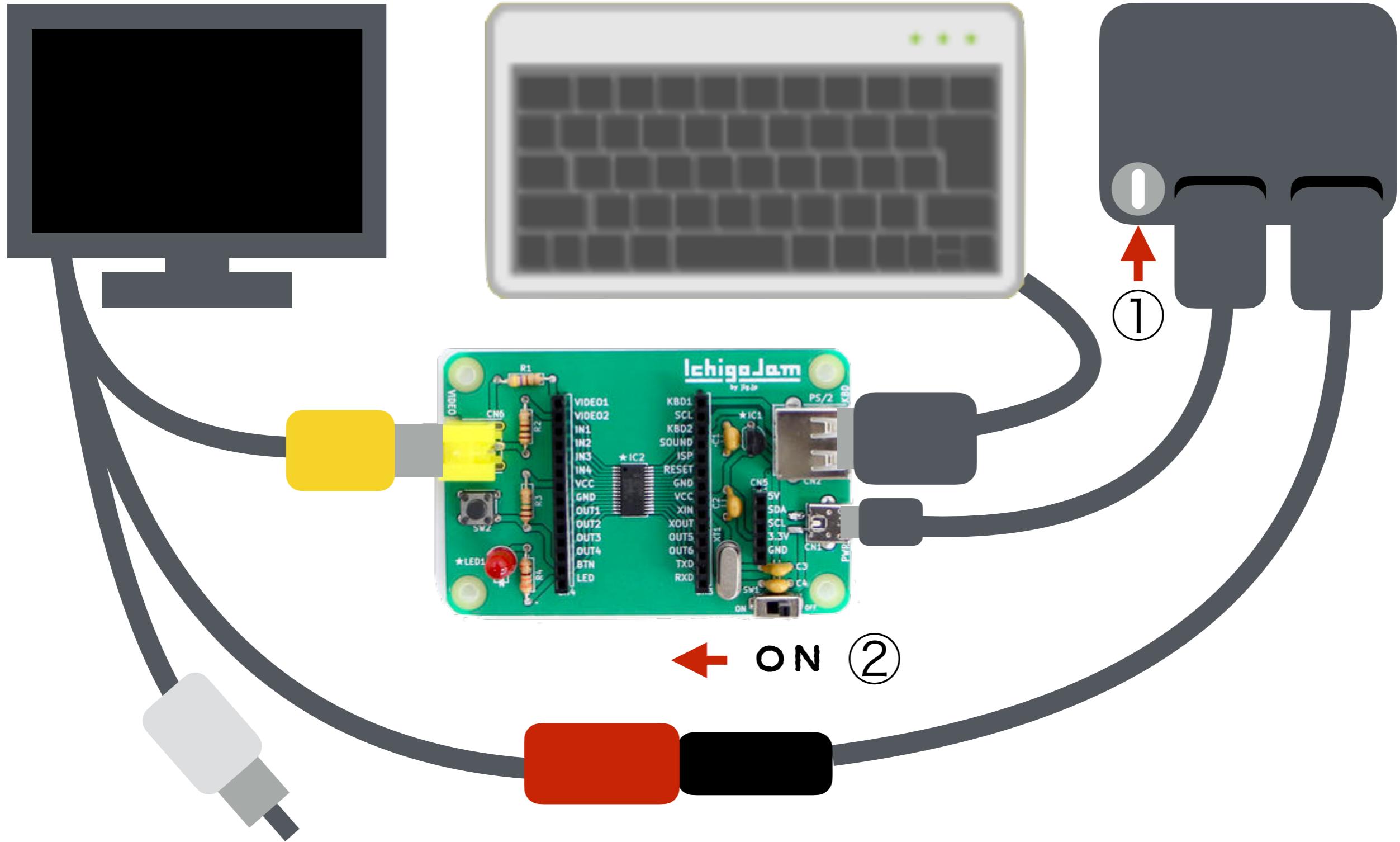
テレビ

キーボード

でんきん



でんげんとIchigoJam をスイッチオン！

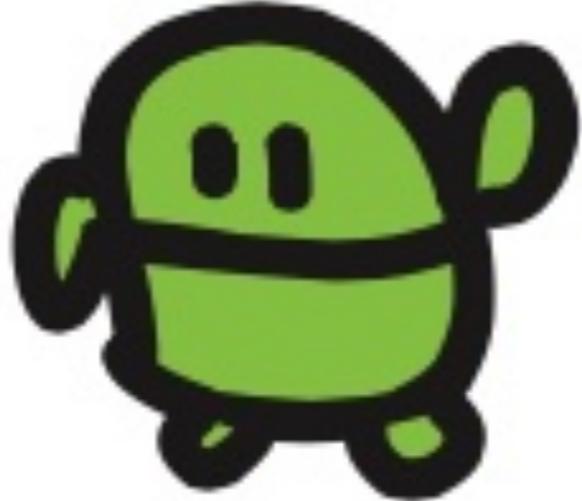


IchigoJam BASIC
OK

てんめつしているのは、カーソル

コンピューターと
はなそう





（ミミ、ナイヨ）

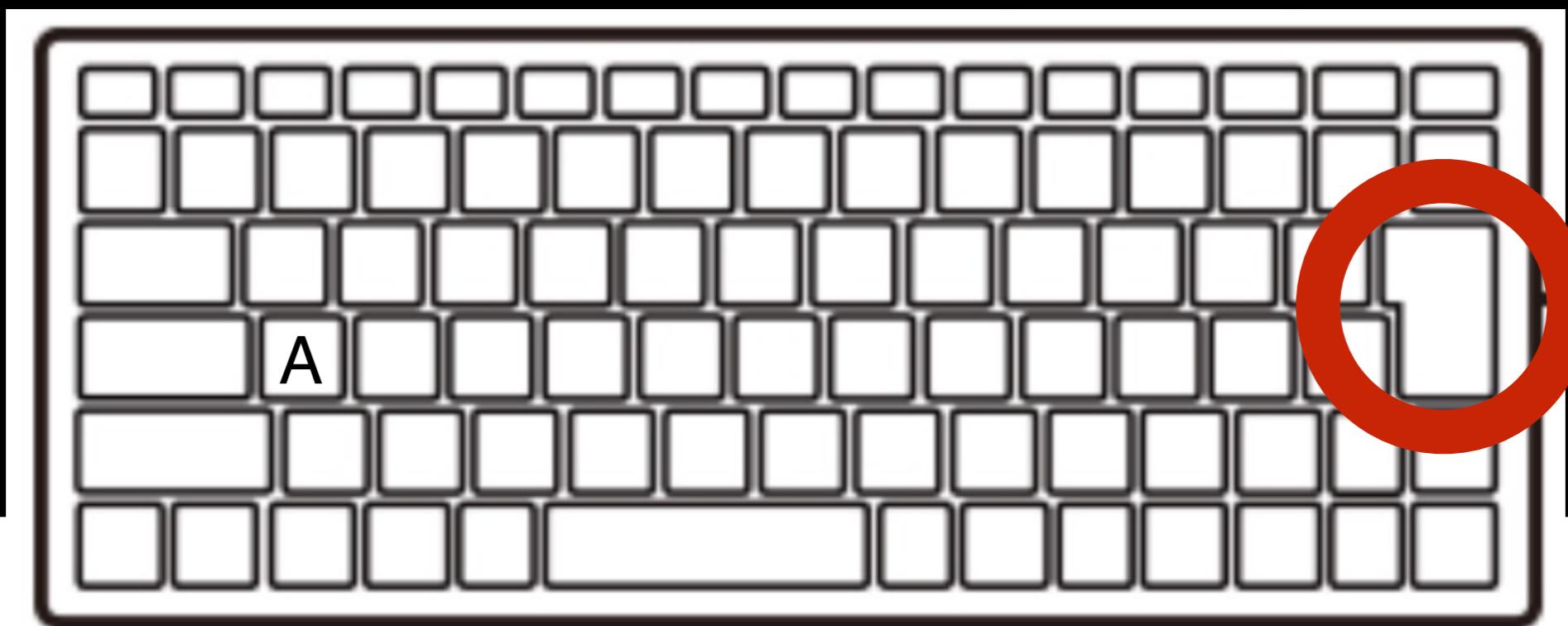
ハローー



IchigoJam BASIC
OK
AI

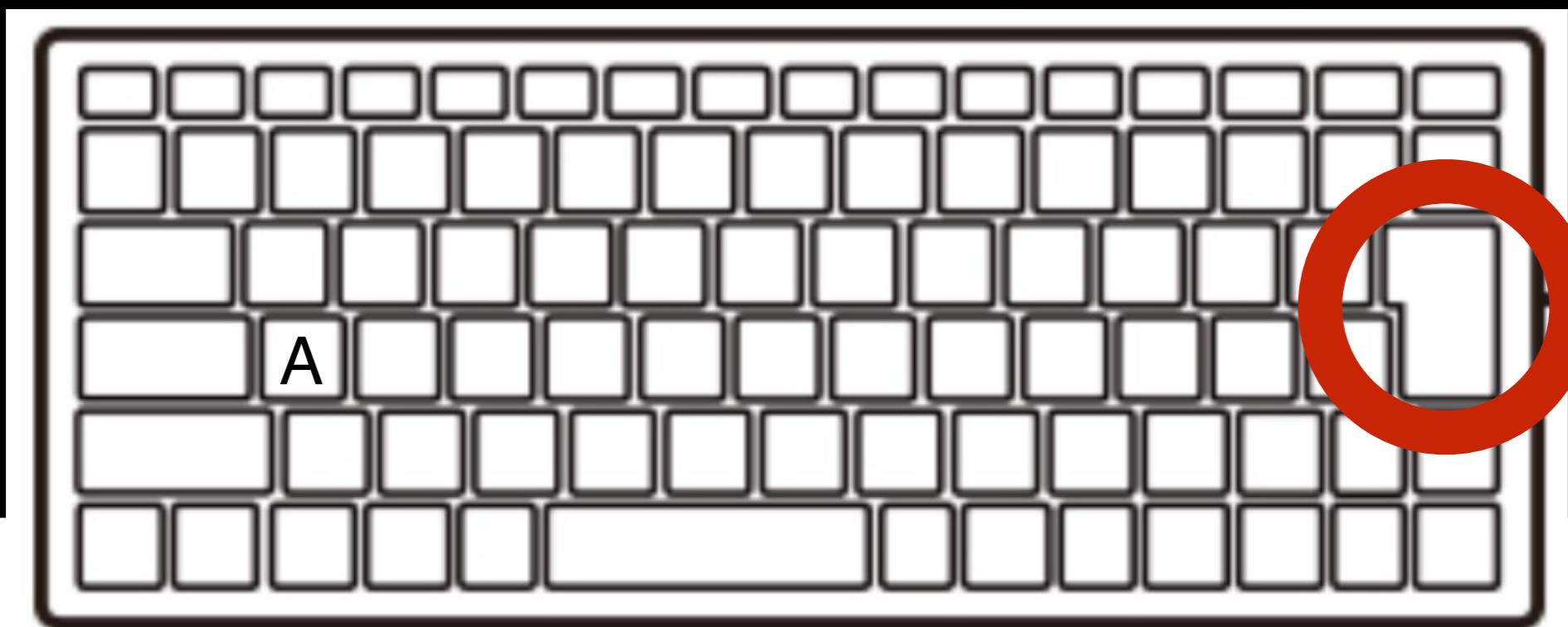
キー ボードで「A」と、うってみよう

I c h i g o J a m B A S I C
OK
AI

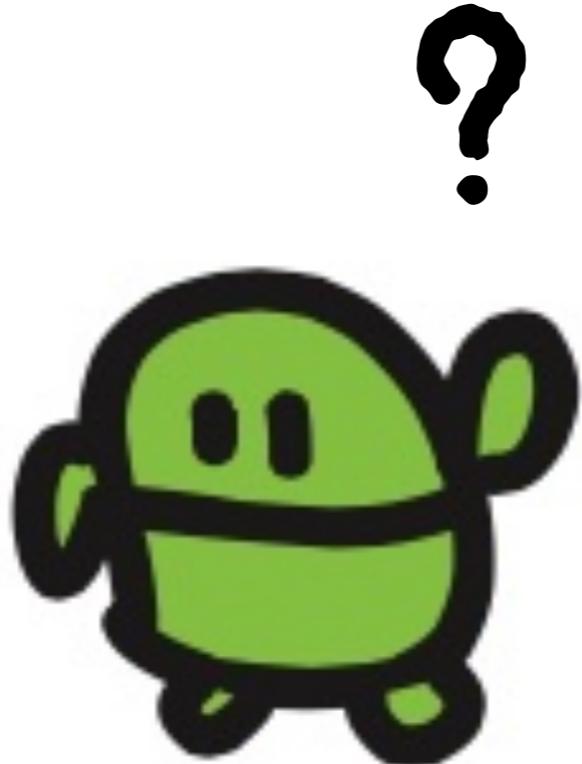


エンターキー

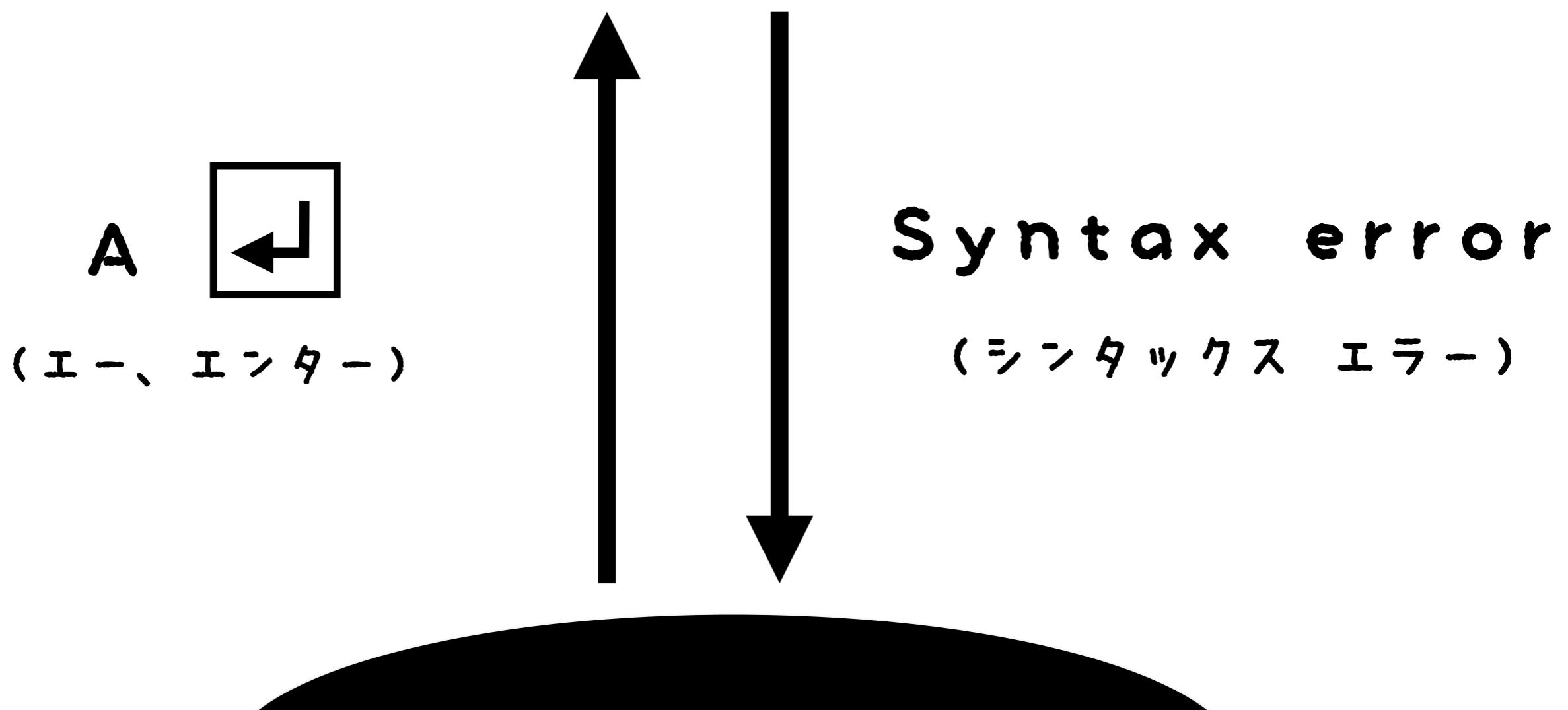
IchigoJam BASIC
OK
A
Syntax error
|

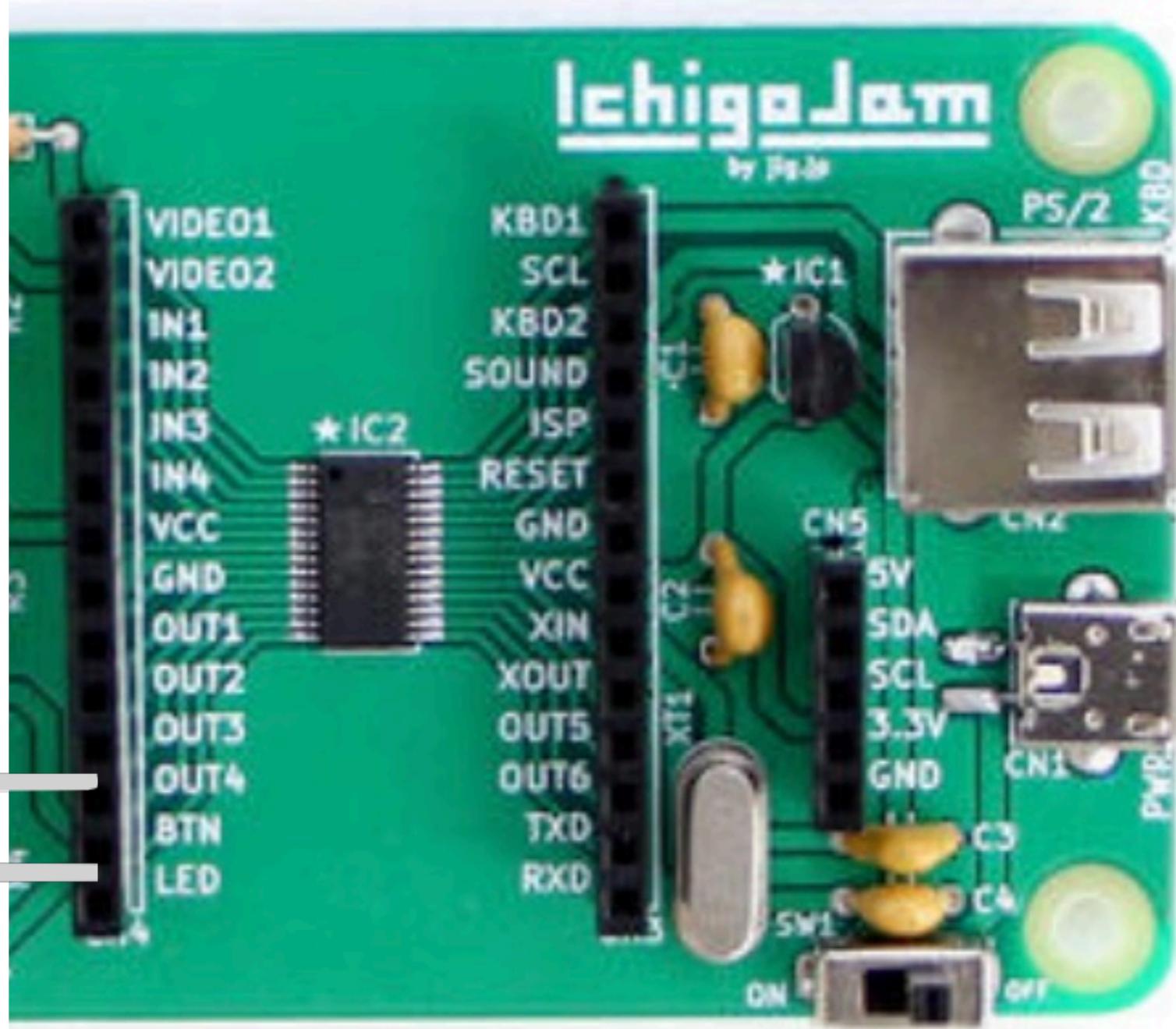
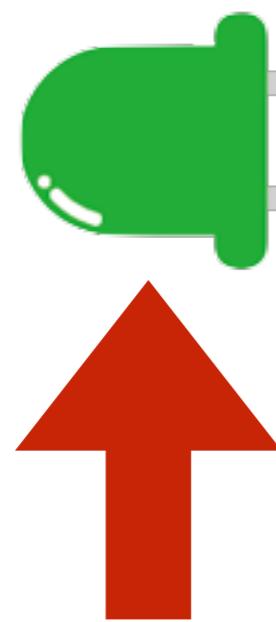


エンターキー



シラナイ
コトバダナー



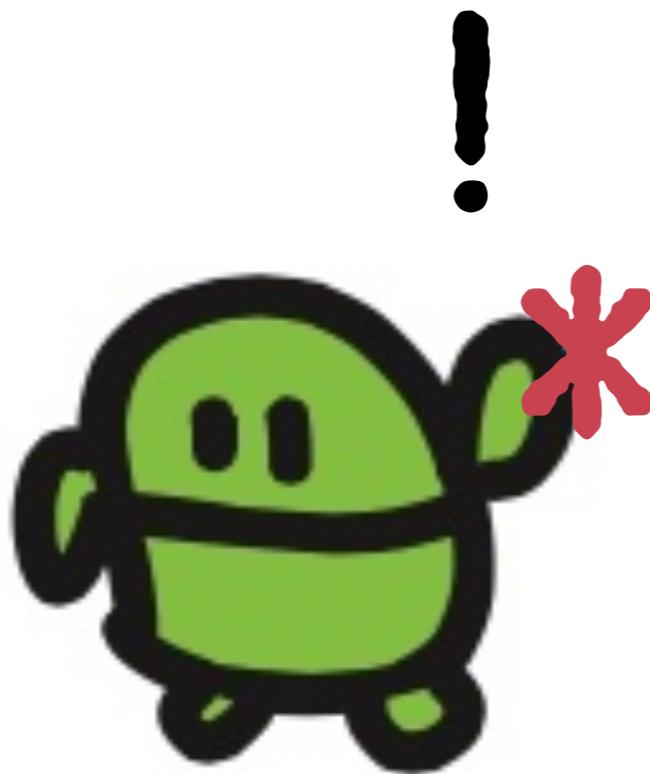


この LED をつけてもらおう

LED1

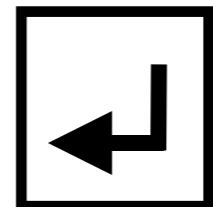


LED1 エンター

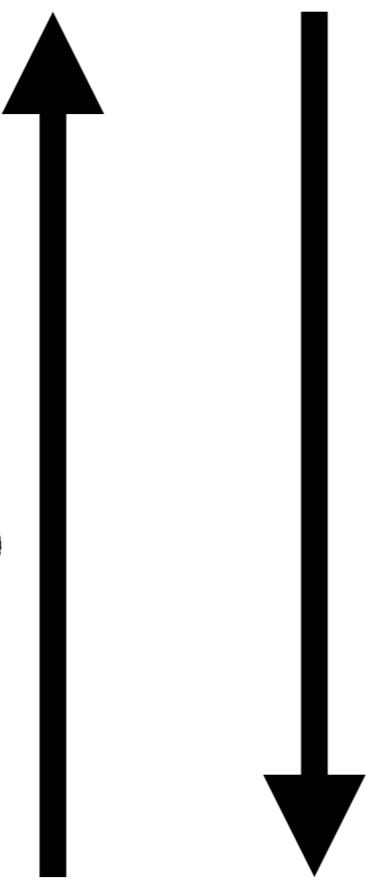


シリテル！

LED1

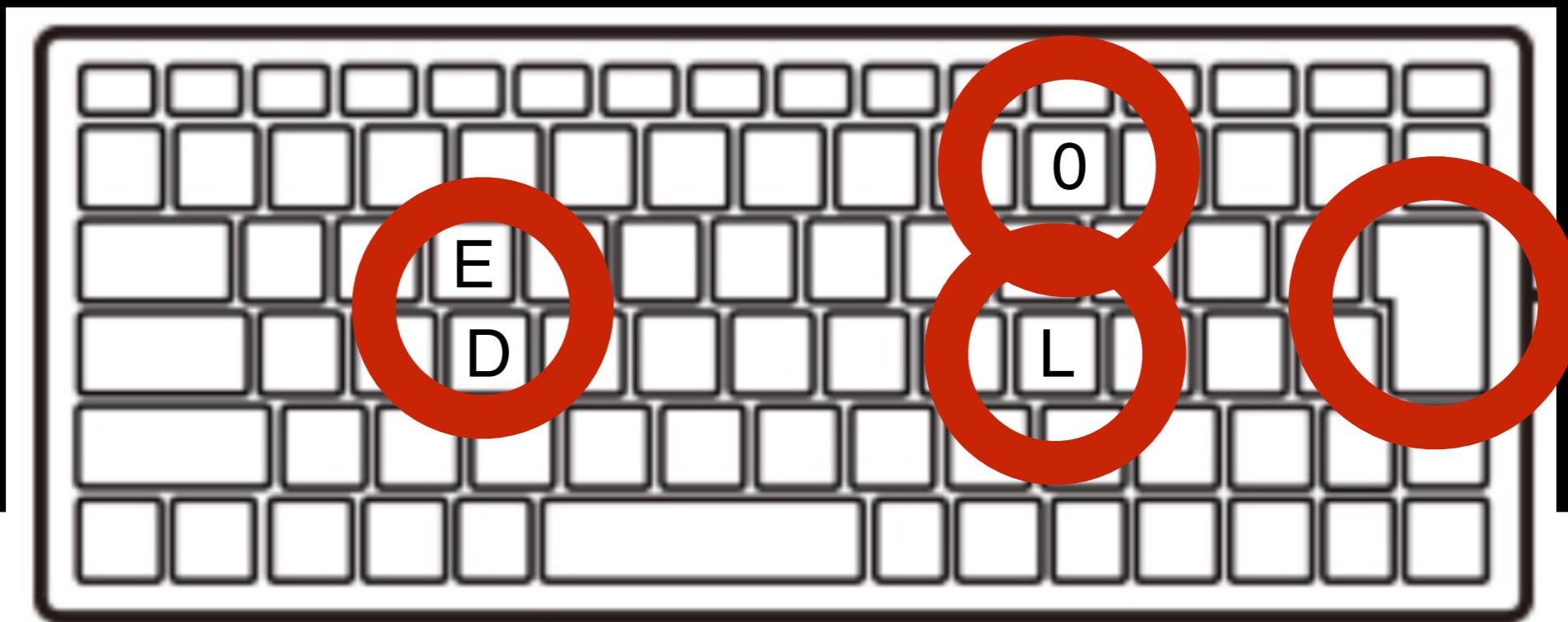


(エルイーディー、ワン、エンター)

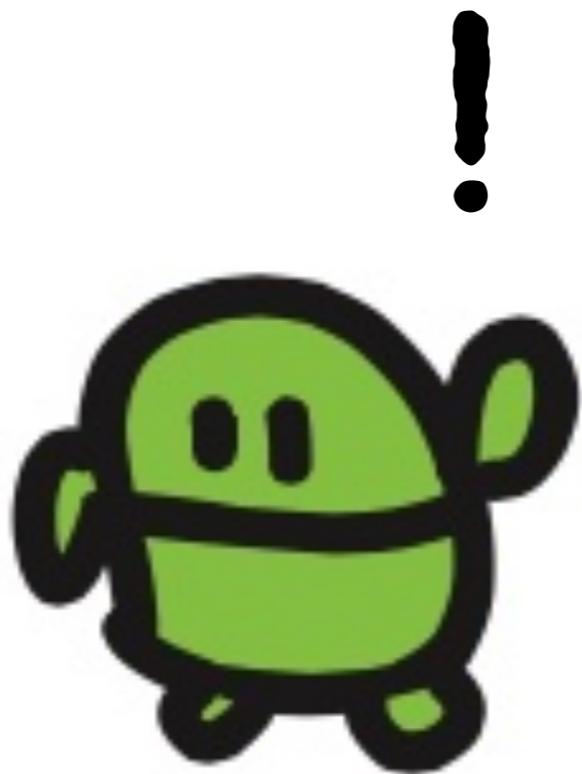


OK
(オーケー)

LEDOI



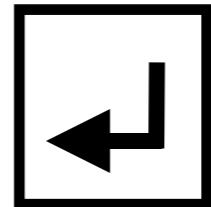
エンターキー



!

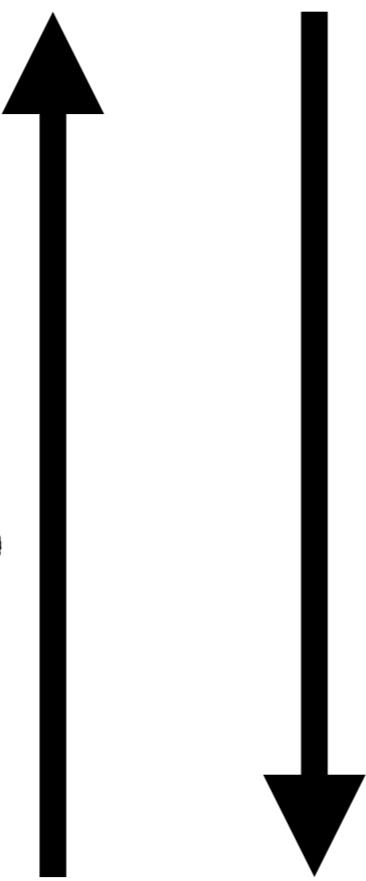
シリテル！

LEDO



(エルイーディー、ゼロ、エンター)

OK

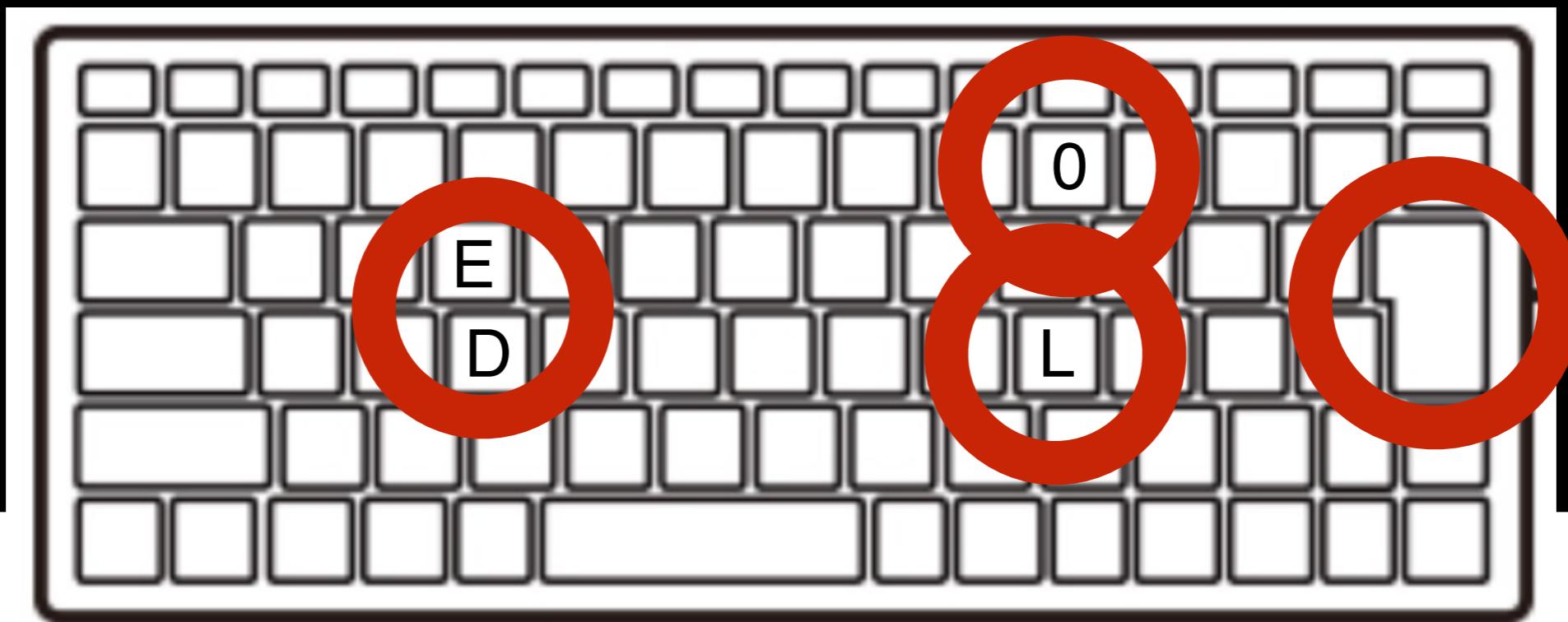


LED1



LED1 エンター

LEDOI

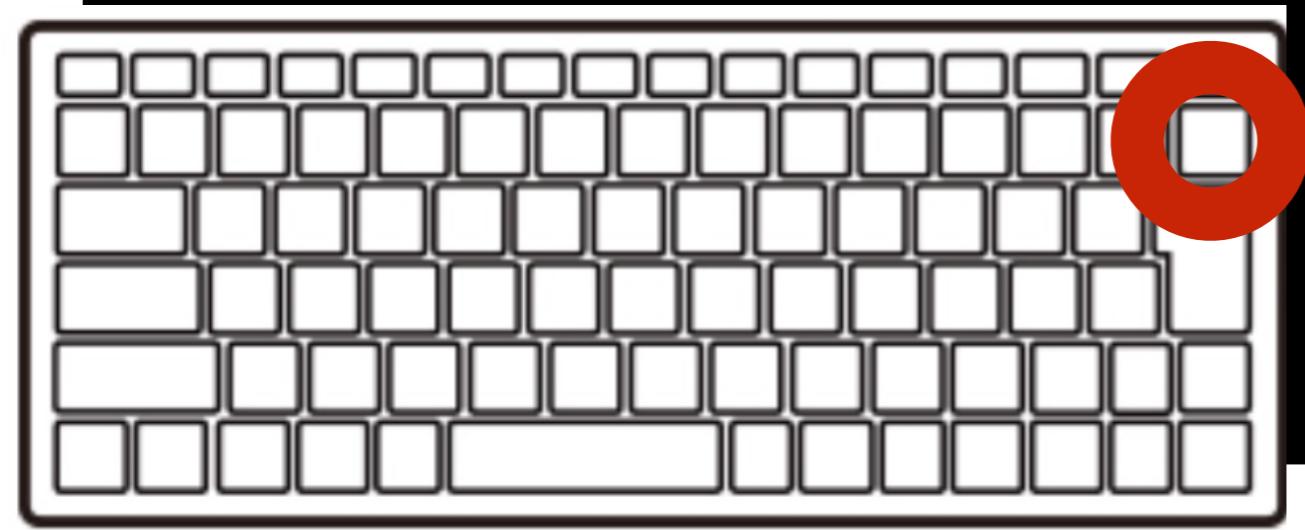


エンターキー

LL LI

うちすぎてみよう

LI



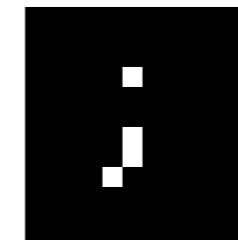
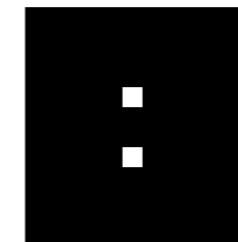
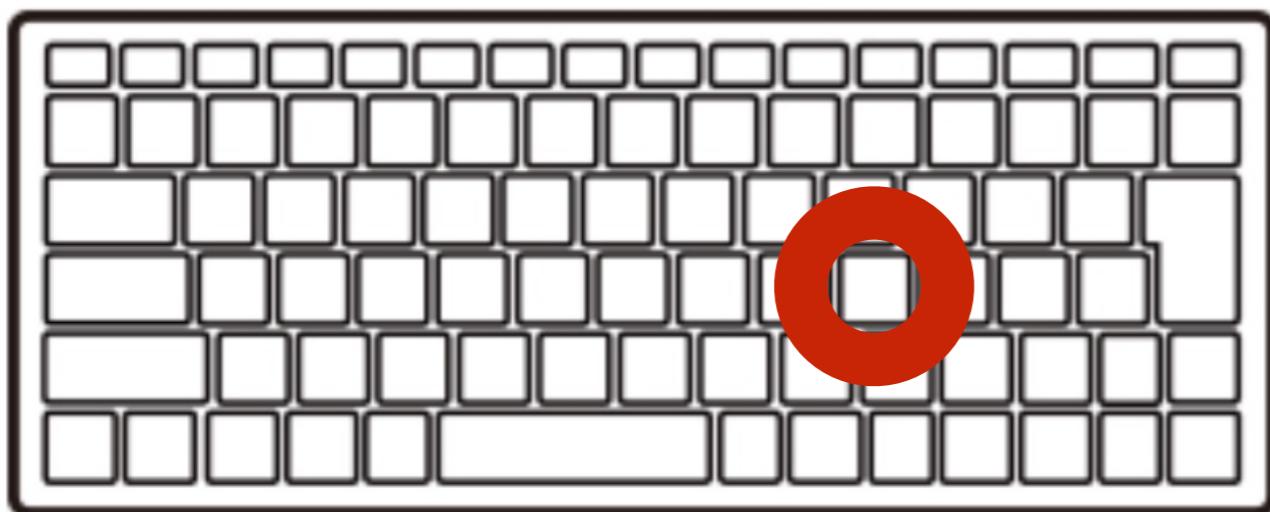
そんなときはバックスペース
(カーソルひだりひとつけて)

ひかるせて、けす！

LED1 : LED0 ↪



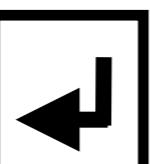
け



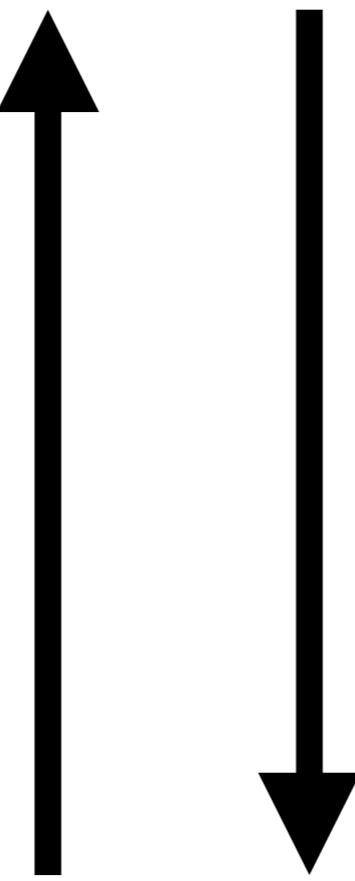
コロン

セミコロン



LED1:LEDO 

(さいごに、エンター)



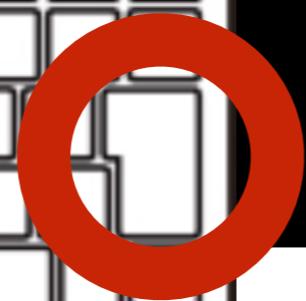
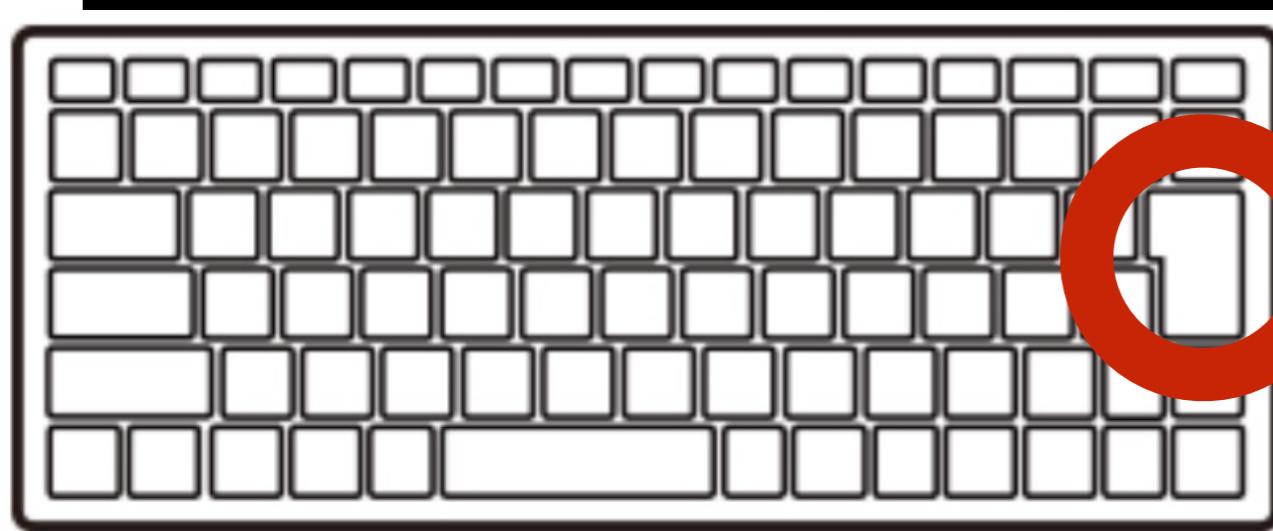
OK

おや？



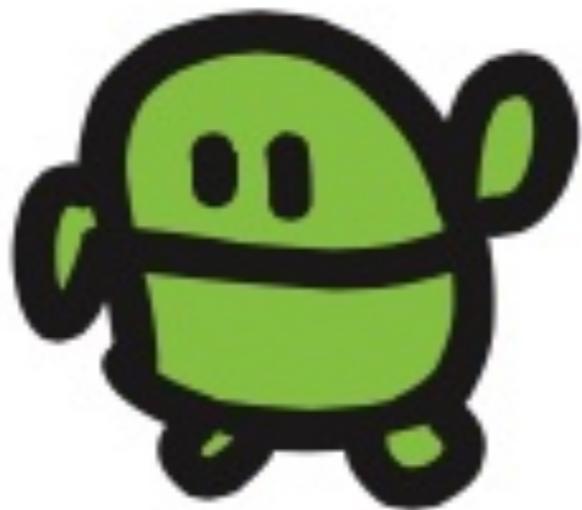
カ - リル 「上」 2 回

LED1 : LED9
OK



エンターでもういちど！

ここで“もんだい”！





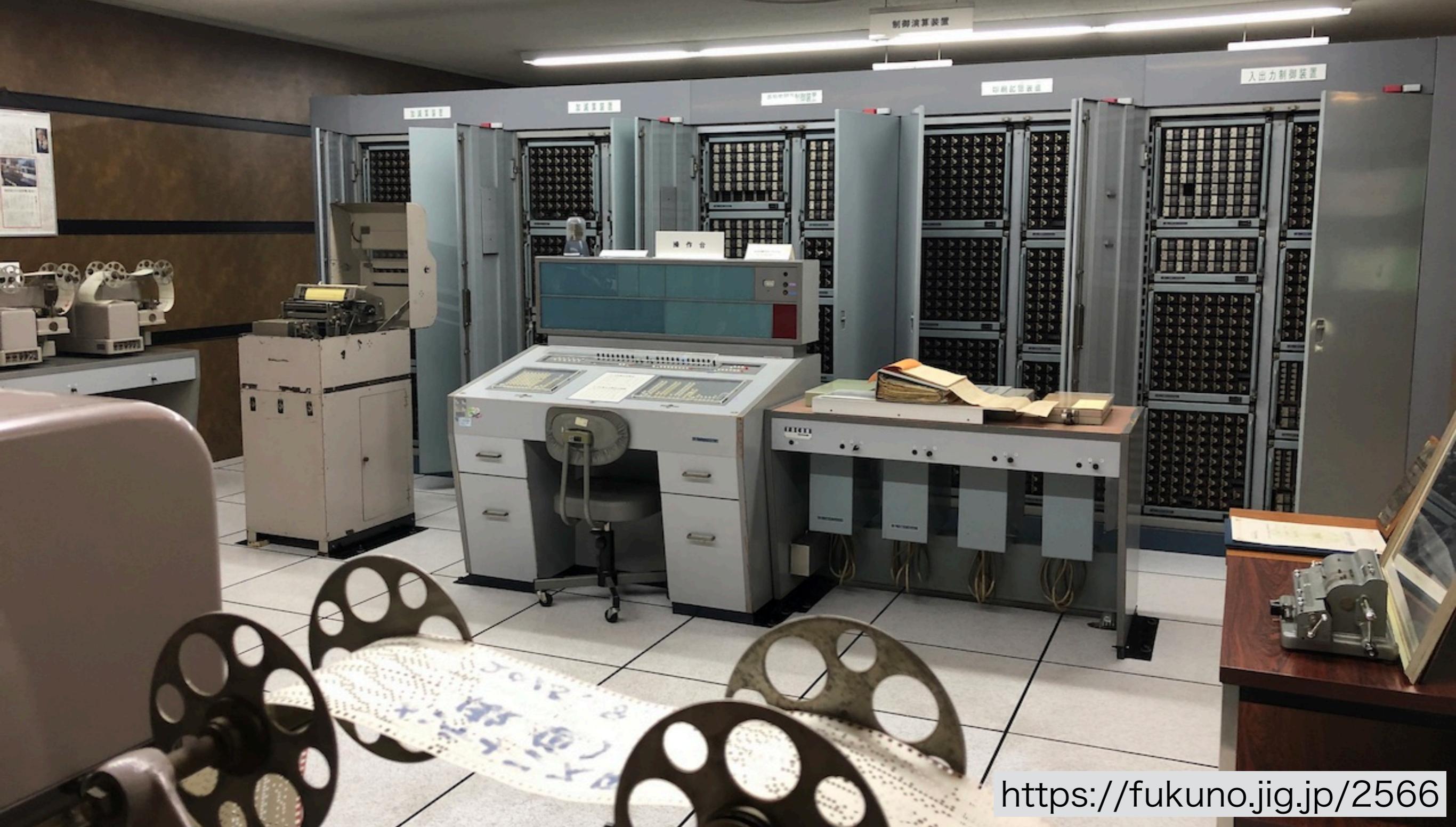
IchigoJam

CPU

100円のコンピューター
1秒間に何回計算できる？

次のページにヒント！

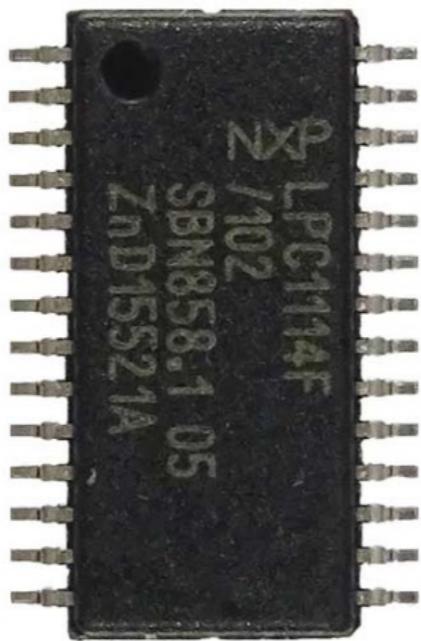
画像、LPC1114 秋月電子



<https://fukuno.jig.jp/2566>

1959年製造 世界最古の動くコンピューター

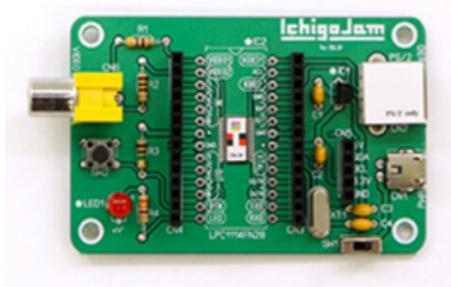
富士通 FACOM128B : 10回/秒



IchigoJam

CPU

1秒に5000万回！



(C)IchigoJam



(C)Apple



(C)NVIDIA



(C)RIKEN

IchigoJam

iPhone 13 Pro

NVIDIA A-100

スパコン富岳

5000万回

15兆回

1200兆回

100京回

IchigoJam
何台分？→

30万台分

2400万台分

200億台分

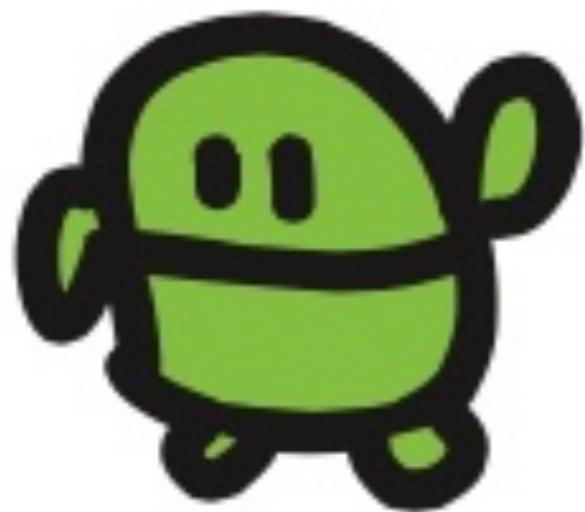
1500円

10万円

200万円

1100億円

まつて = WAIT

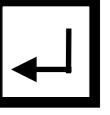


まって

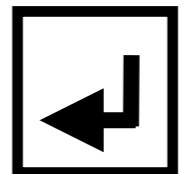
W A I T 1 8 0 ↵

エンター、おしてから
OKとかえるまでなんびよう？

ひかって、3びょうまって、けして

LED1 : WAIT180 : LED0 

うしろにつづけてかいて、エンター
2かいてんめっ！

LED1:WAIT100:LED0:WAIT10
:LED1:WAIT10:LED0 

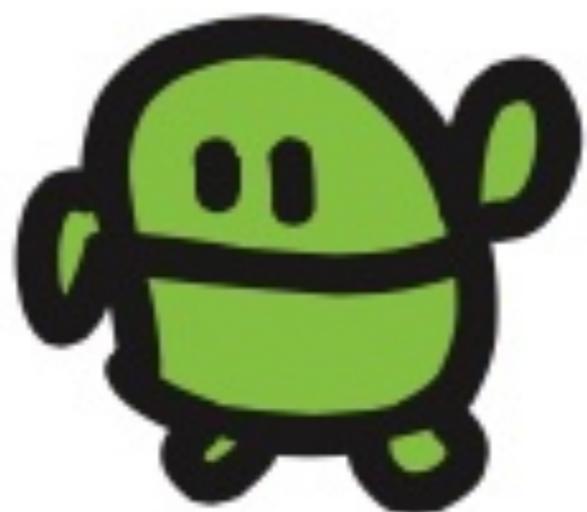
2かい、ひかった？

10かいひからせるには？



10回ひかるはずw

プログラム

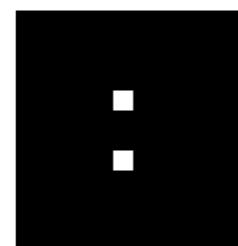


1 LED1 : WAIT10 ←

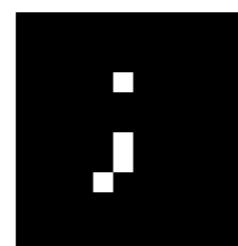
2 LED0 : WAIT10 ←

スペース

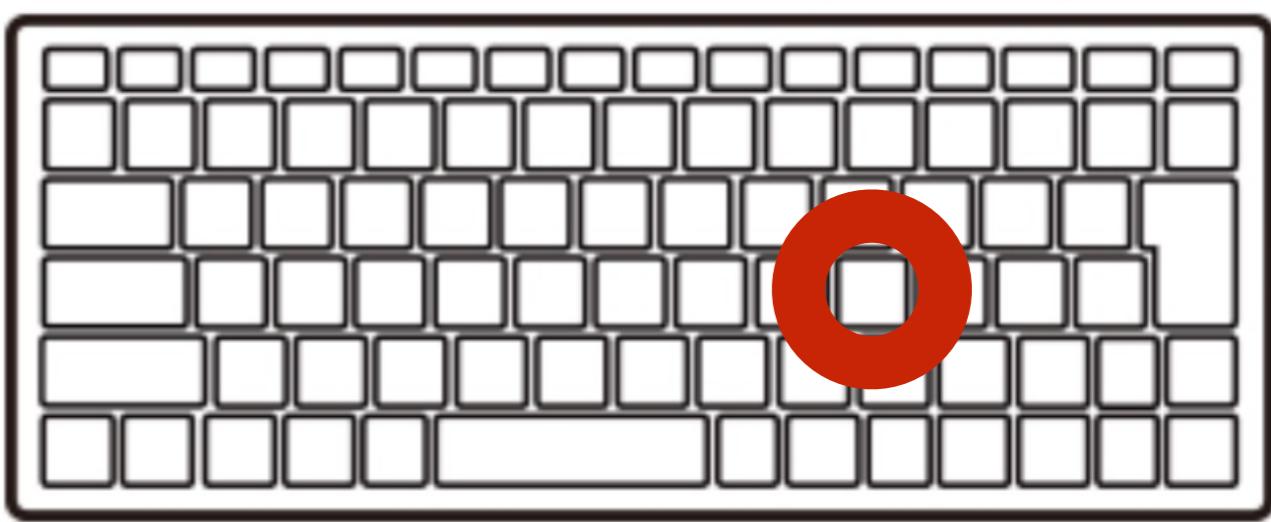
け



コロン



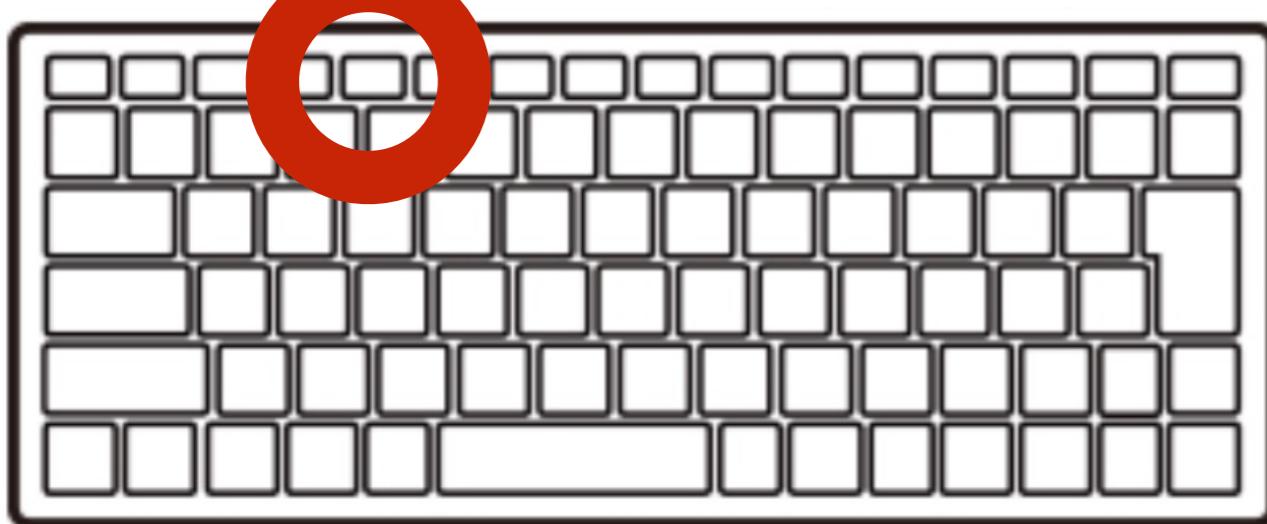
セミコロン



リスト（プログラムみせて）

LIST

F4



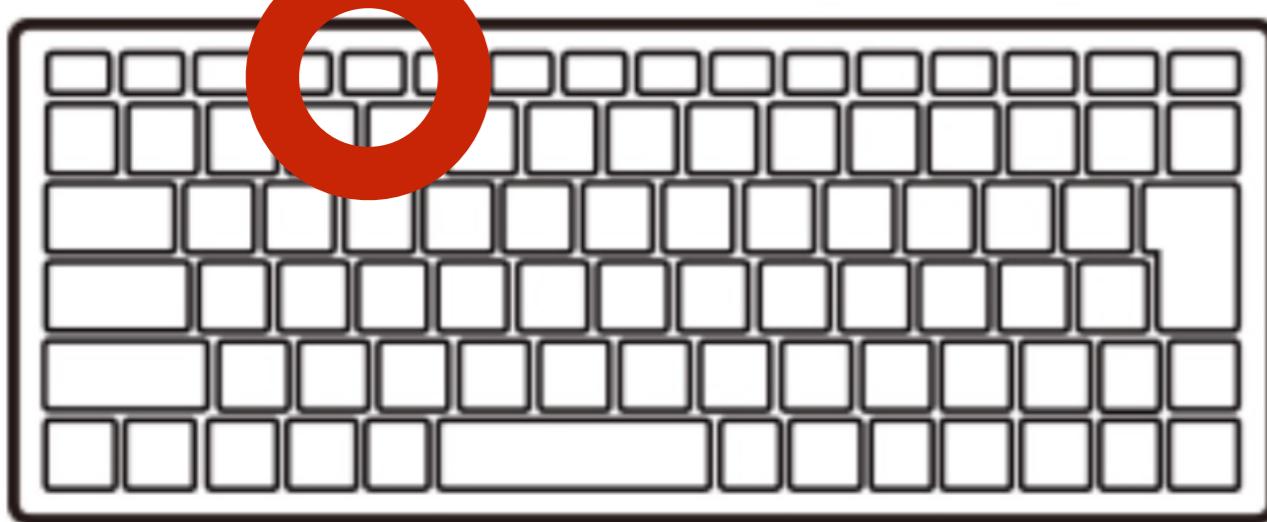
おぼえてるよ



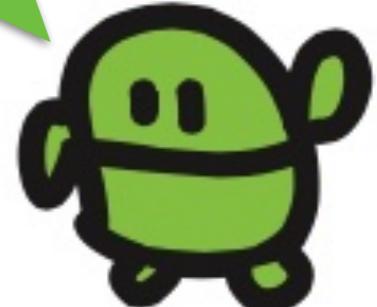
ラン（はしれ！／うごかす）

RUN

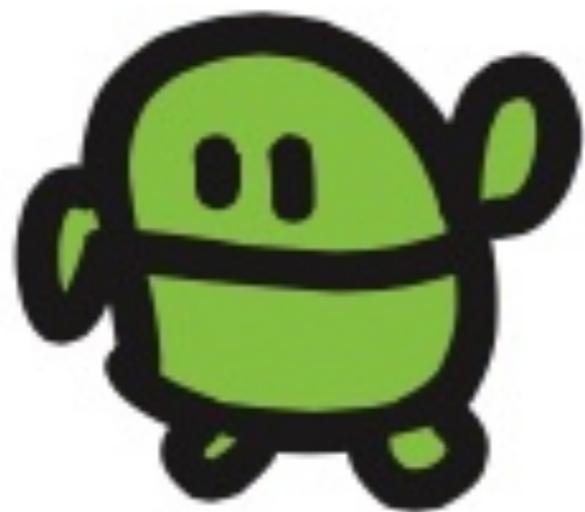
F5



F5



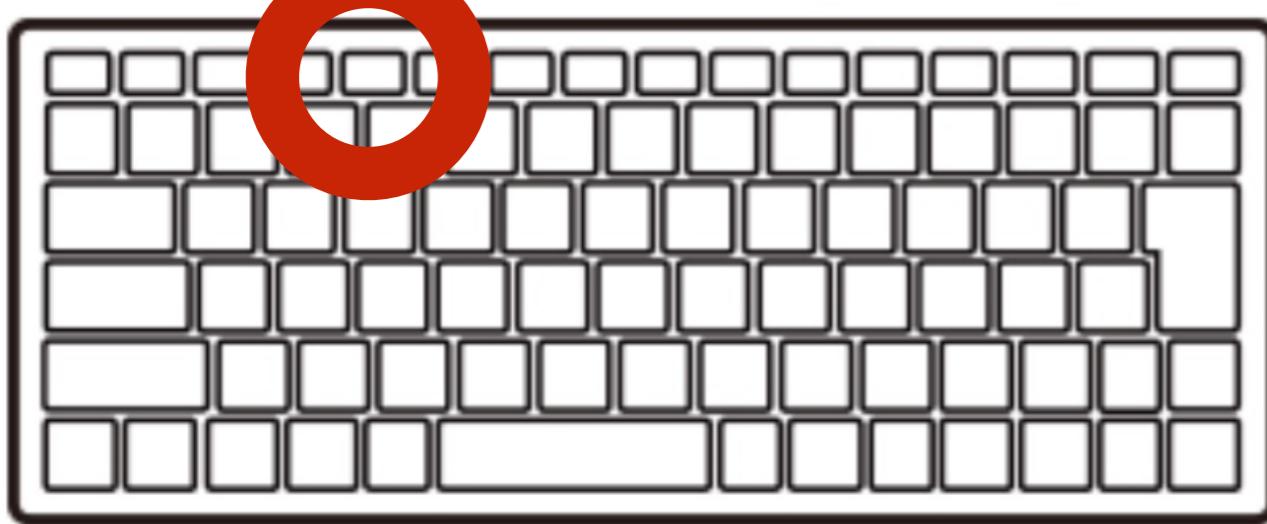
1000回やって？



くりかえし

3 GOT01 ↵

F5



いつまで？

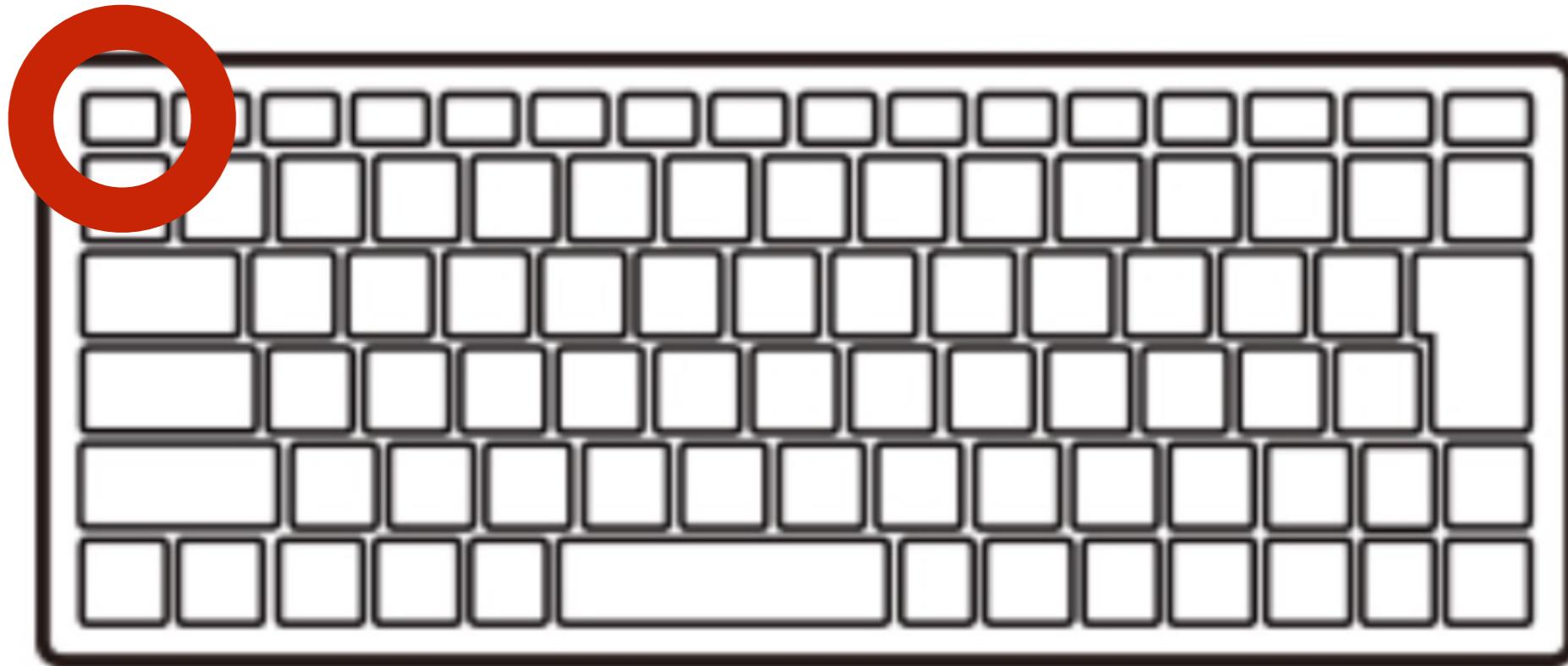
エルチカケーム

とめてひかってたら、かち！



とまって！エスケープキー

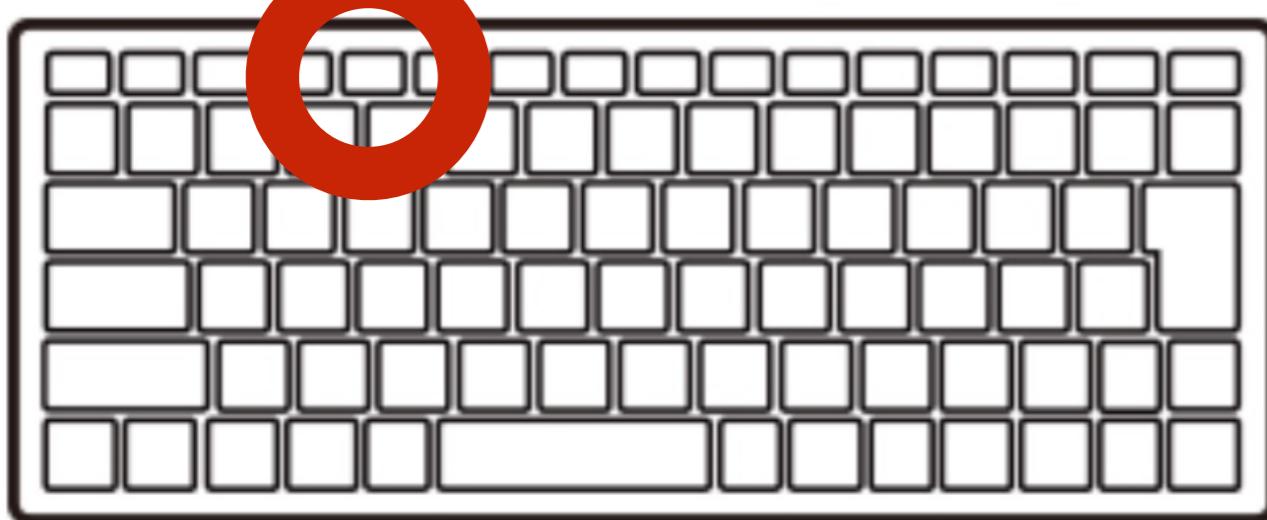
[ESC] + -



リスト（プログラムみせて）

LIST

F4



おぼえてるよ

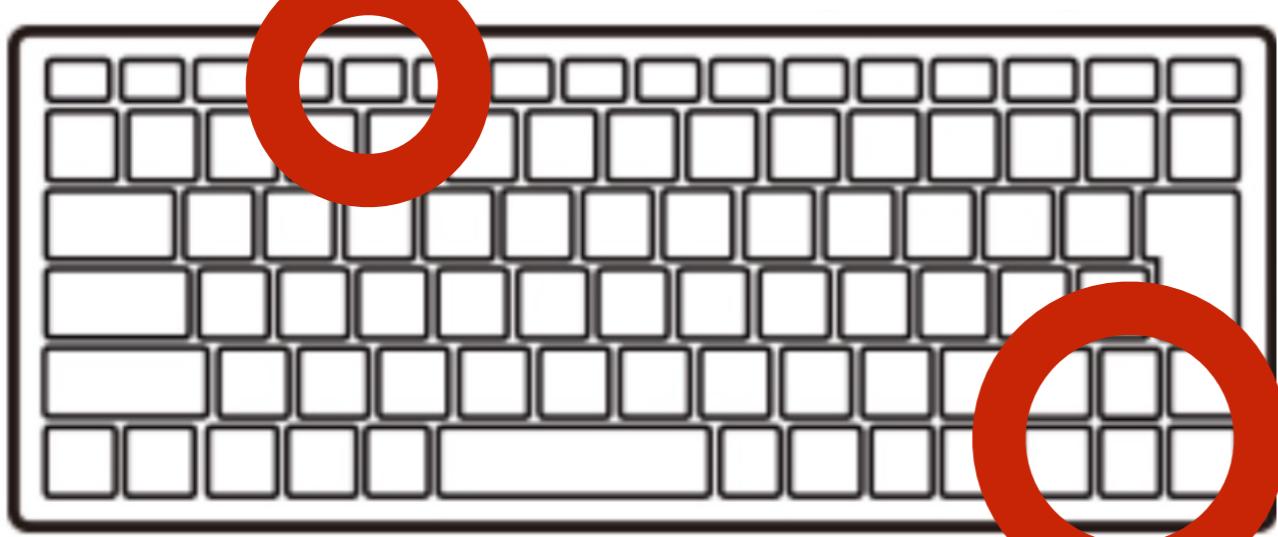


カーソルキーとバックスペースで“かいぞう
かえたぎょう”で「エンター」をおして「F5」

```
1 LED1 : WAIT10
2 LED0 : WAIT30 ←
3 GOTO1
```

F5

カーソルキー

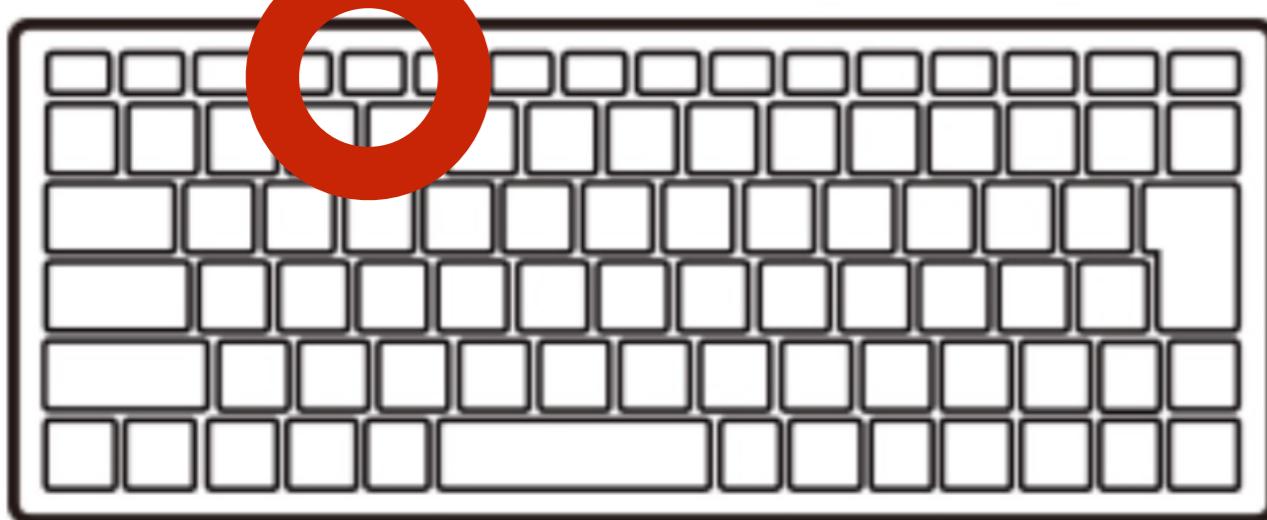


じゅうじざい？

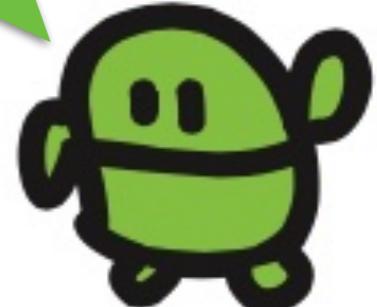
リスト（プログラムみせて）

LIST

F4



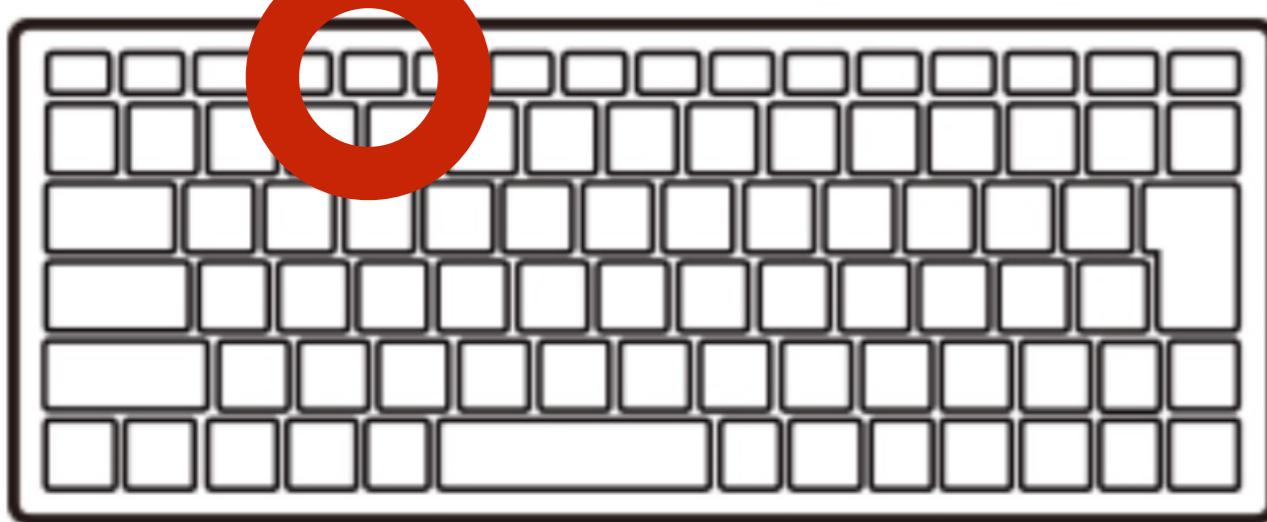
かわった！



ラン（はしれ！／うごかす）

RUN

F5



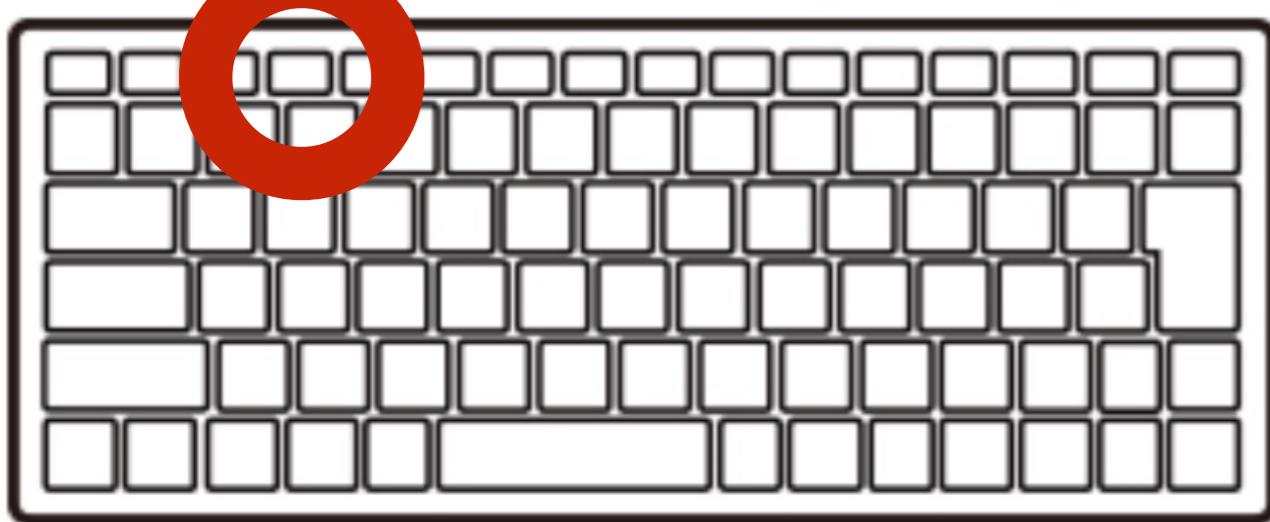
かわった！



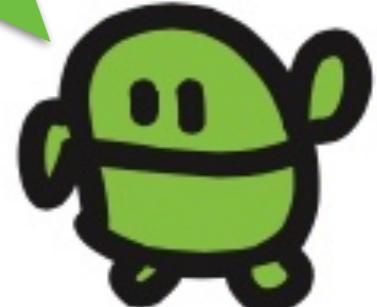
ほぞん（プログラム書き込み）

SAVE

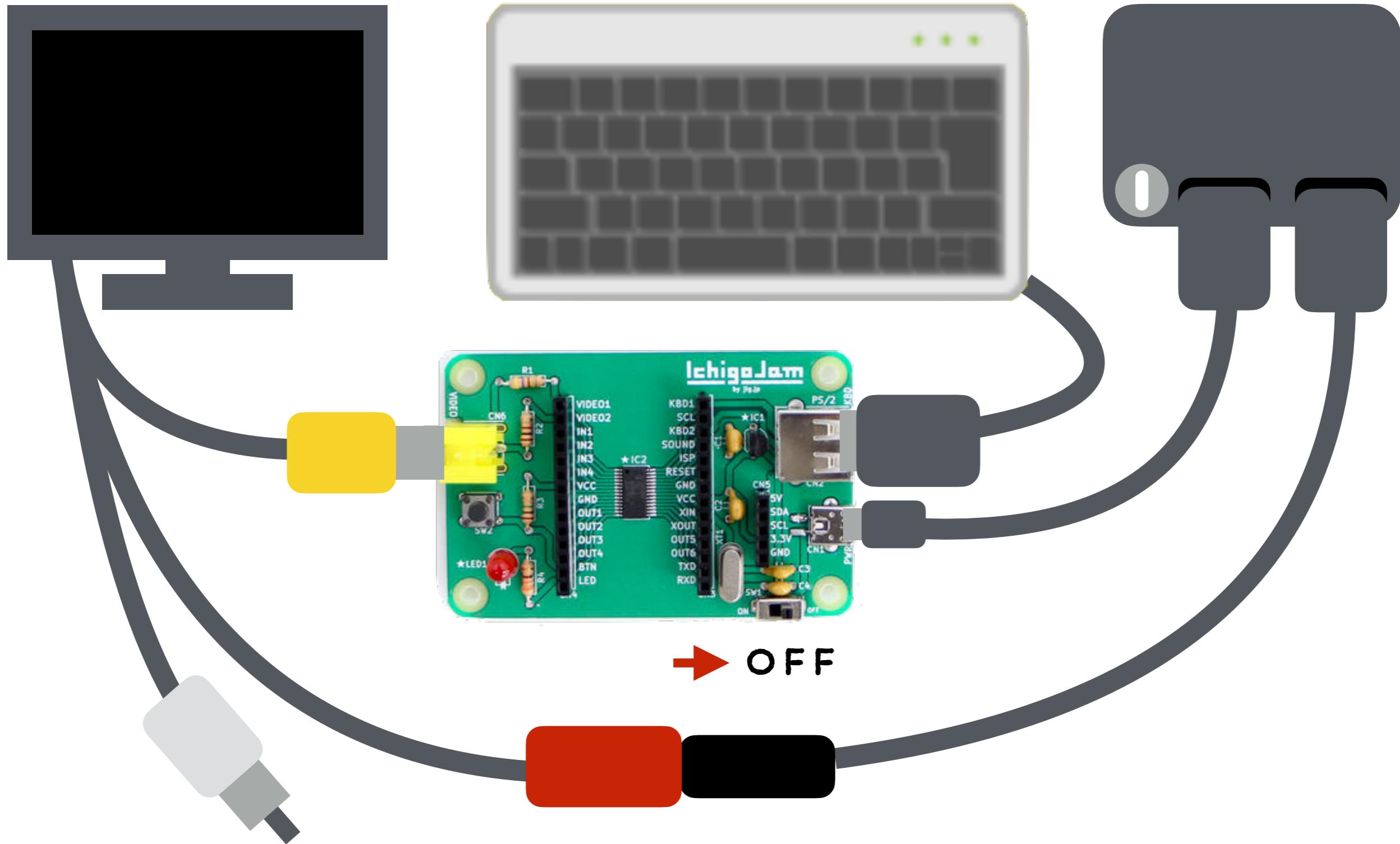
F3



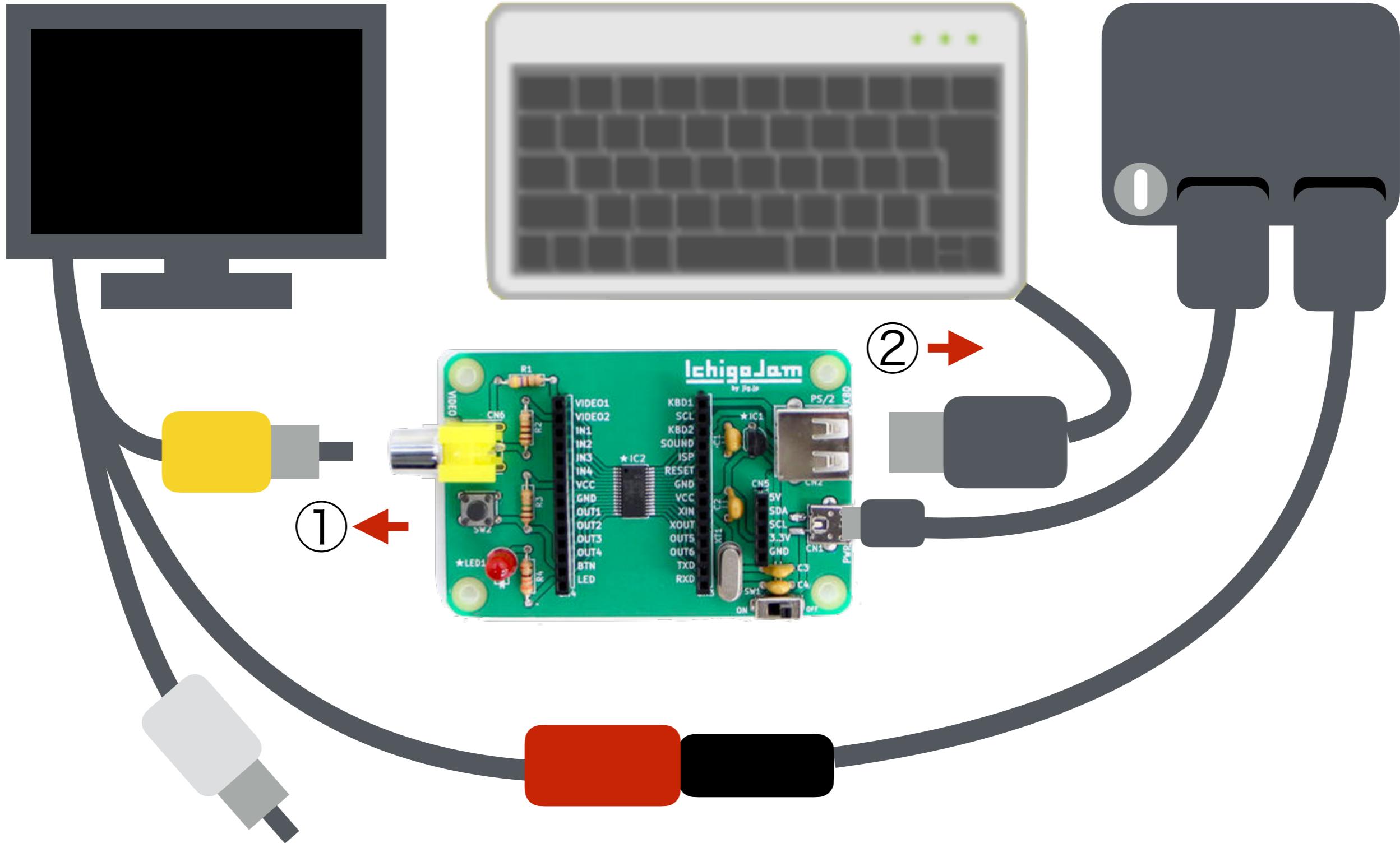
F3、0、エンター



IchigoJam のスイッチ、オフ

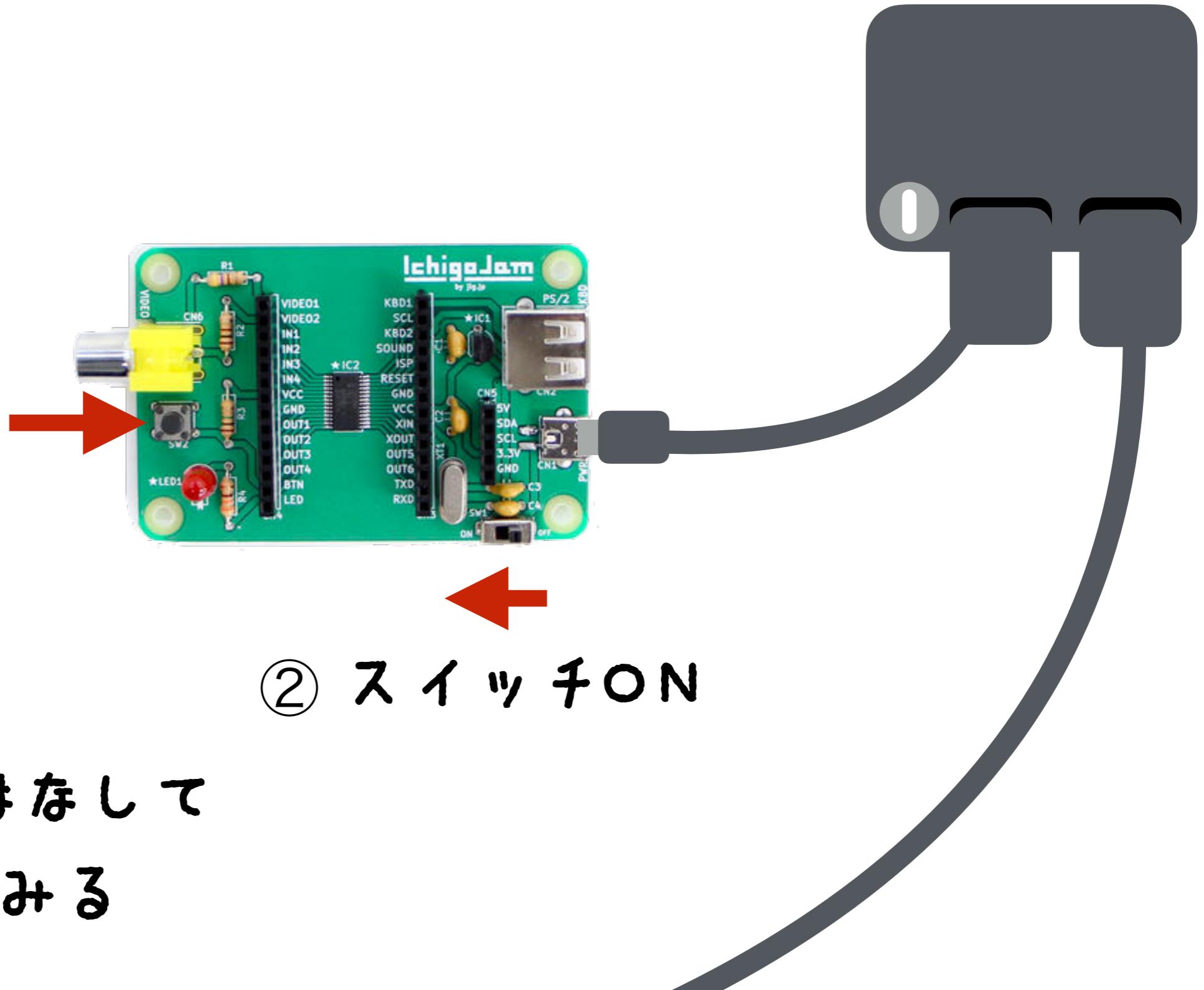


テレビとキーボードをぬこう



ボタンをおしながらスイッチオン！

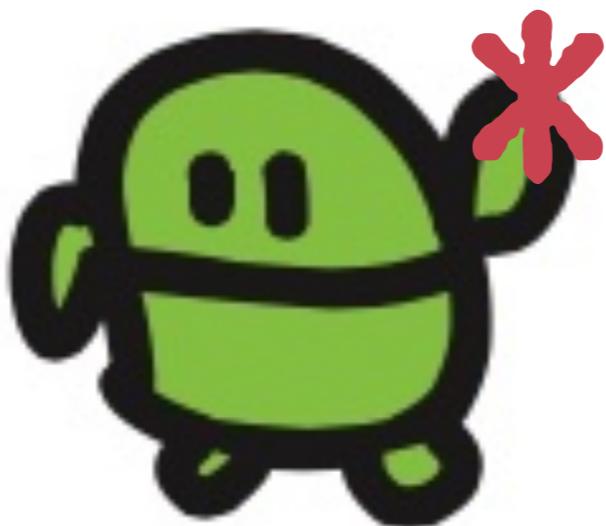
① ボタンを
おしながら



③ ボタンをはなして
LEDを見る

エルチカラボット

IoTの「T」できだ！



みのまわりのロボット



パナソニック洗濯機

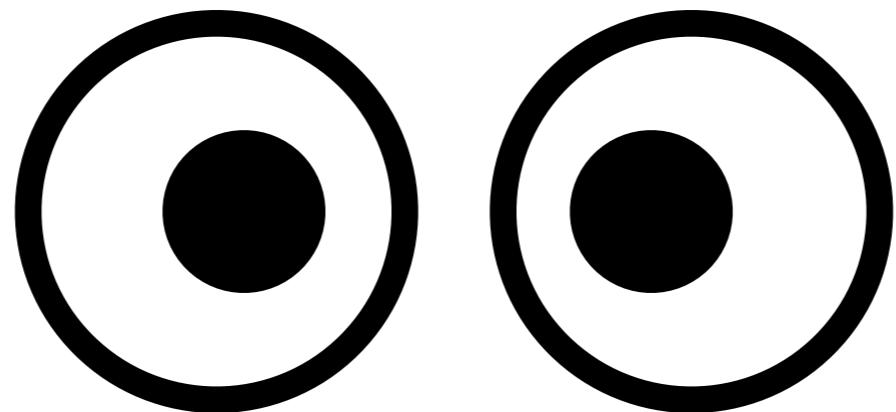


ぜんぶ、だれかが
プログラミングしたもの

コンピューターは
どこにいる？

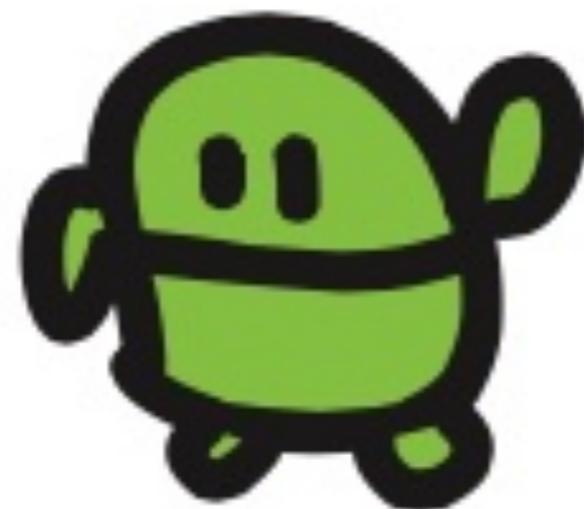


お家のコンピューター
さがしてみよう！

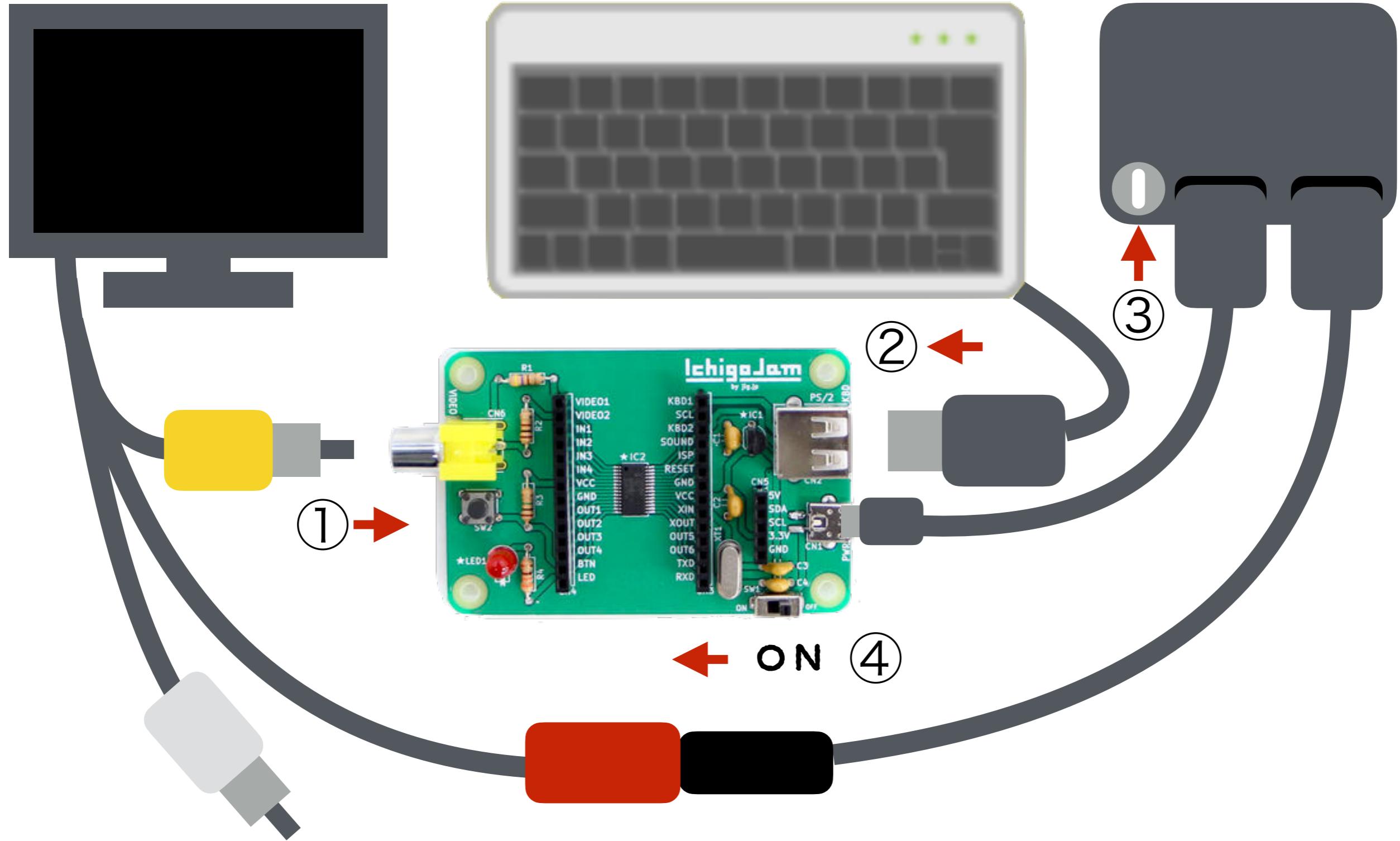


ケ" - ムでまなぶ

プログラミング



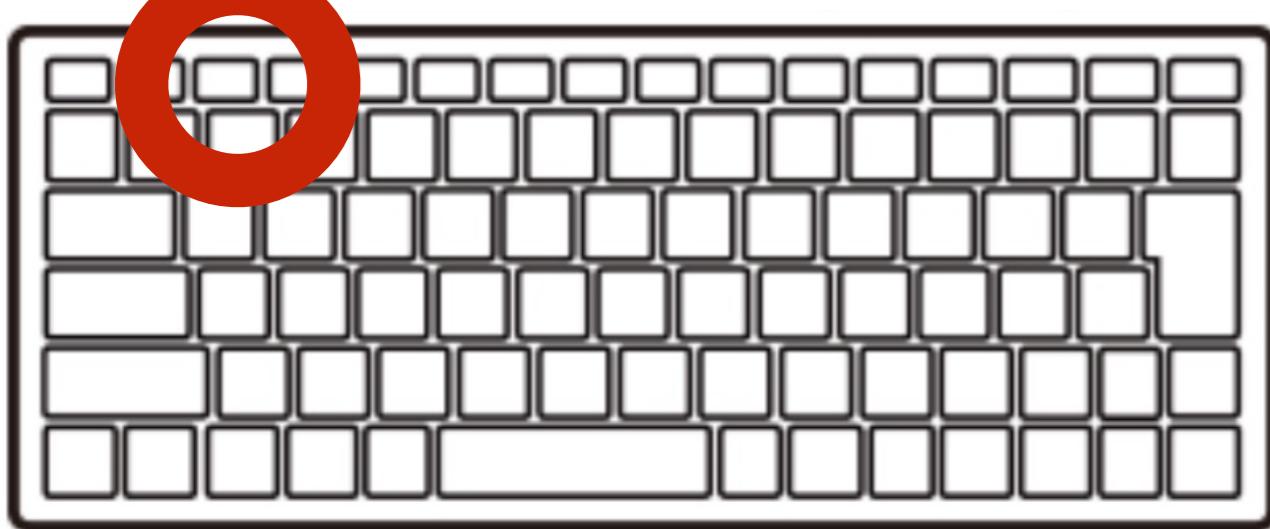
テレビとキーボードをつなないで、ON!



よみこんでみよう

LOAD ↵

F2



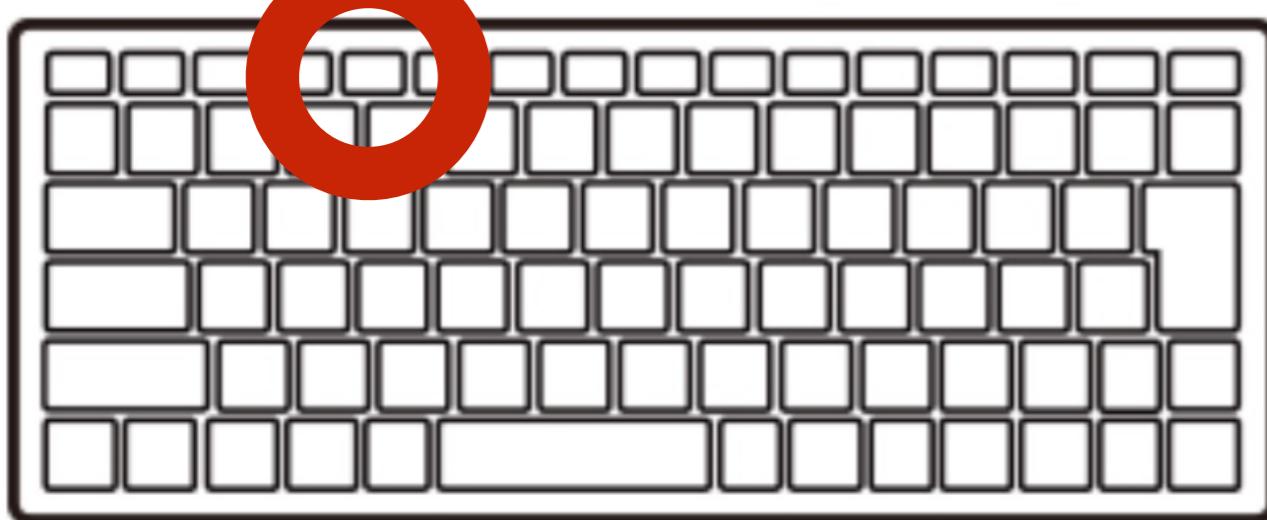
F2、エンター



リスト（プログラムみせて）

LIST

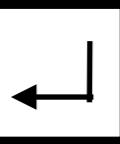
F4



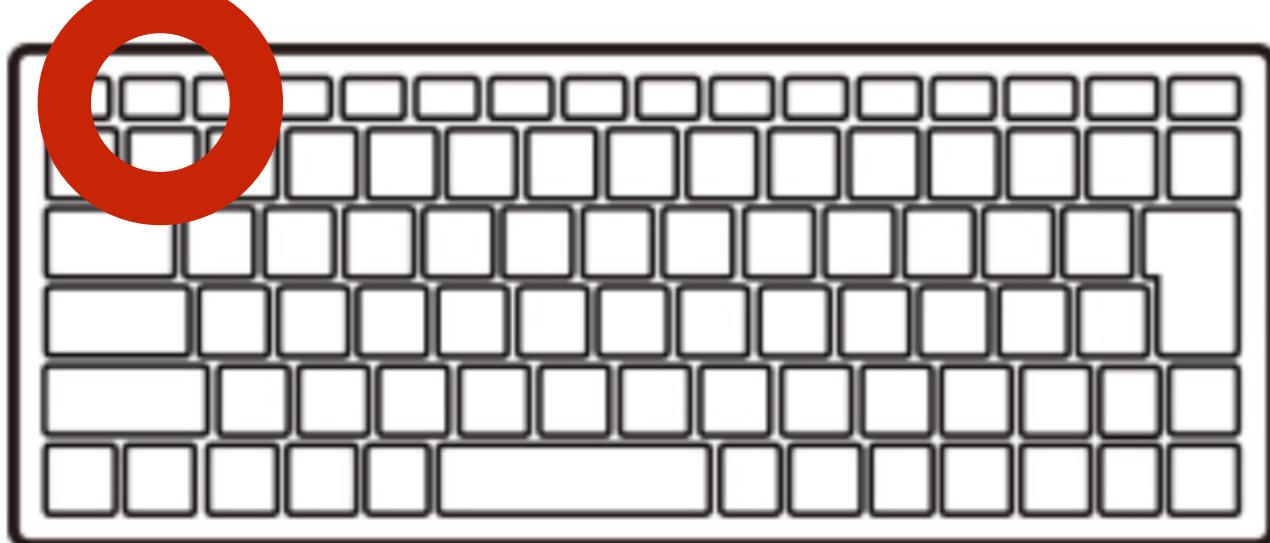
おもいだしたよ



がめんをきれいに

CLS 

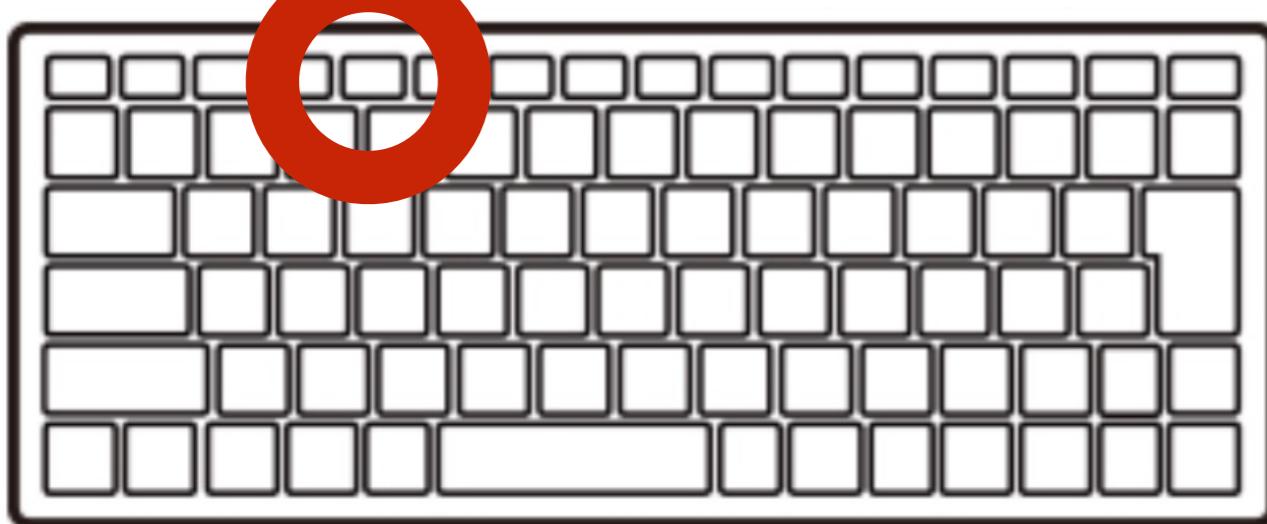
F1



リスト（プログラムみせて）

LIST

F4



おもいだしたよ



さいしょから（プログラムクリア）

NEW 

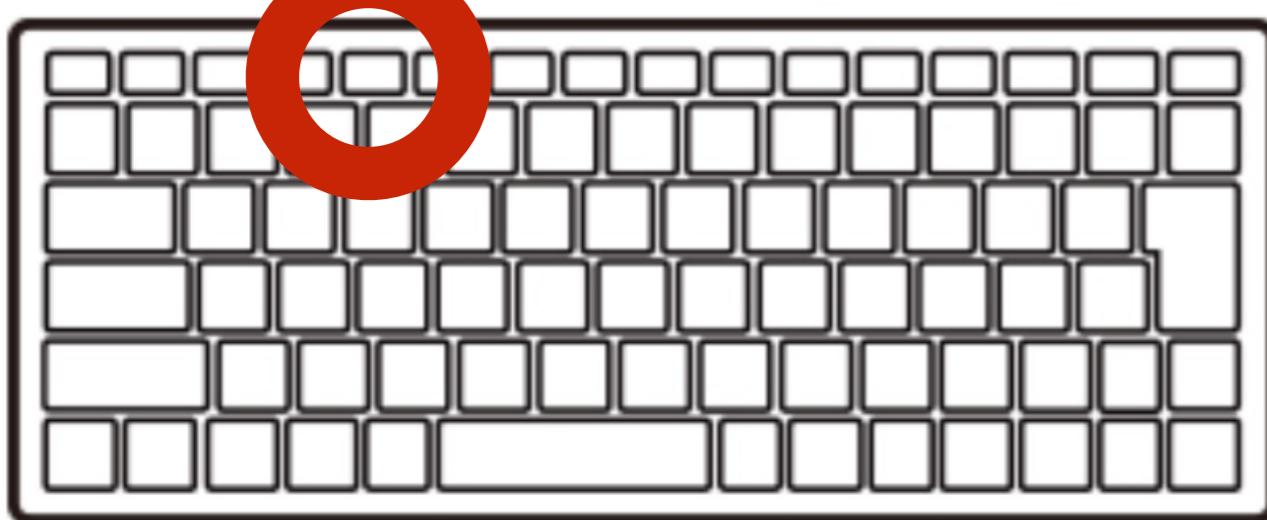
ほぞんしたのは
きえないよ



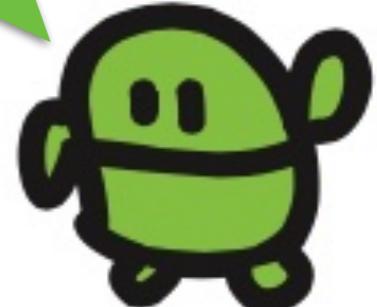
リスト（プログラムみせて）

LIST

F4



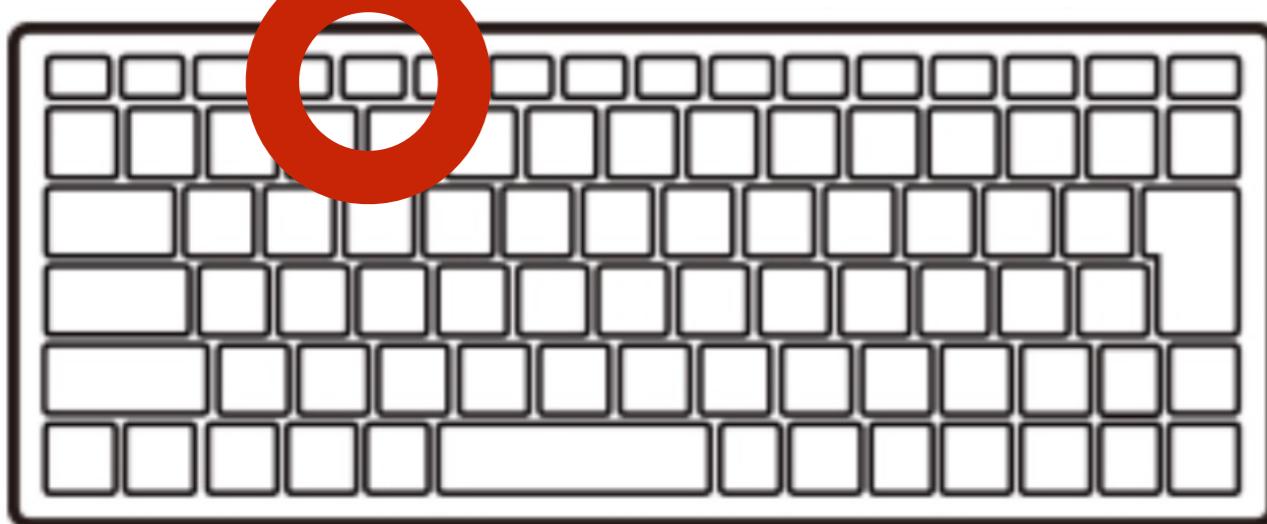
わすれたよ



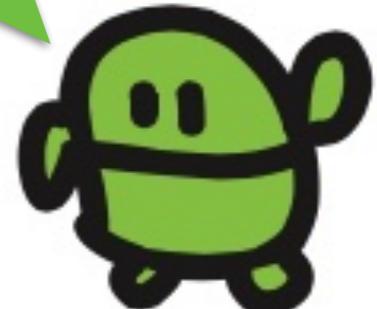
うごかして

RUN

F5

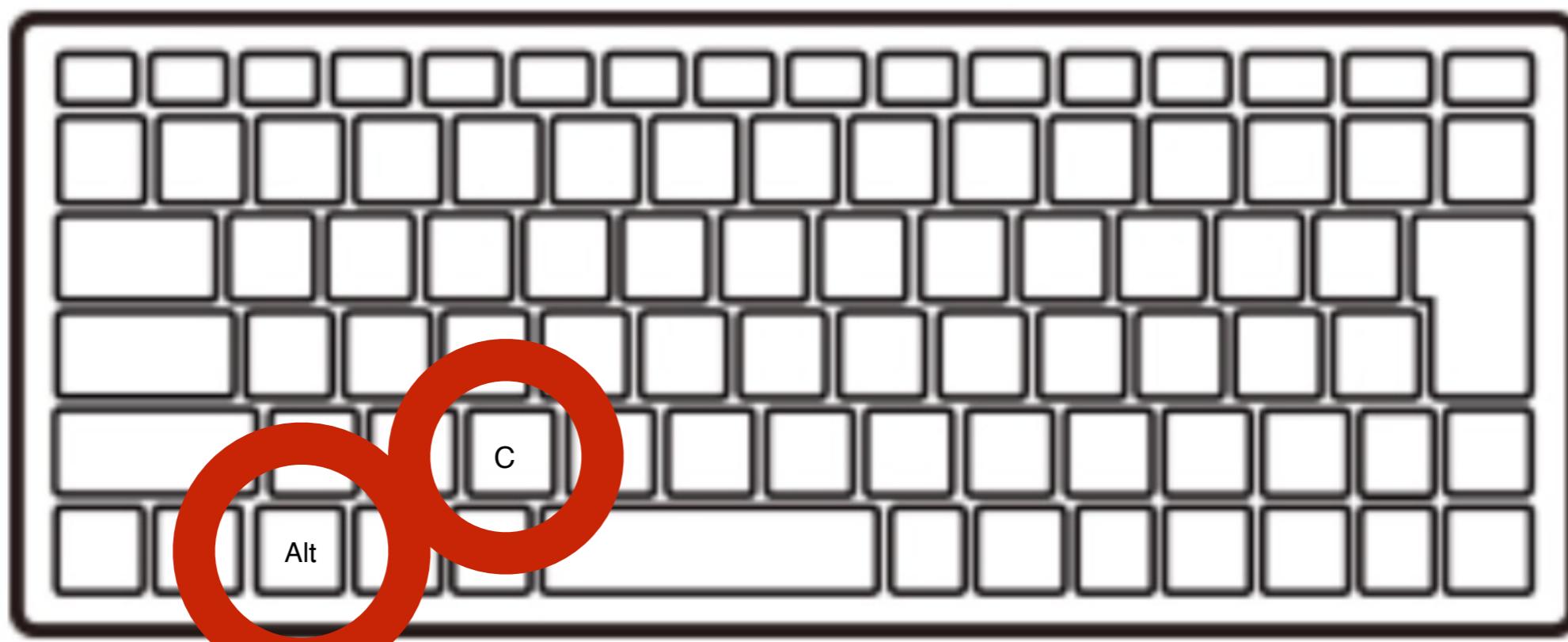
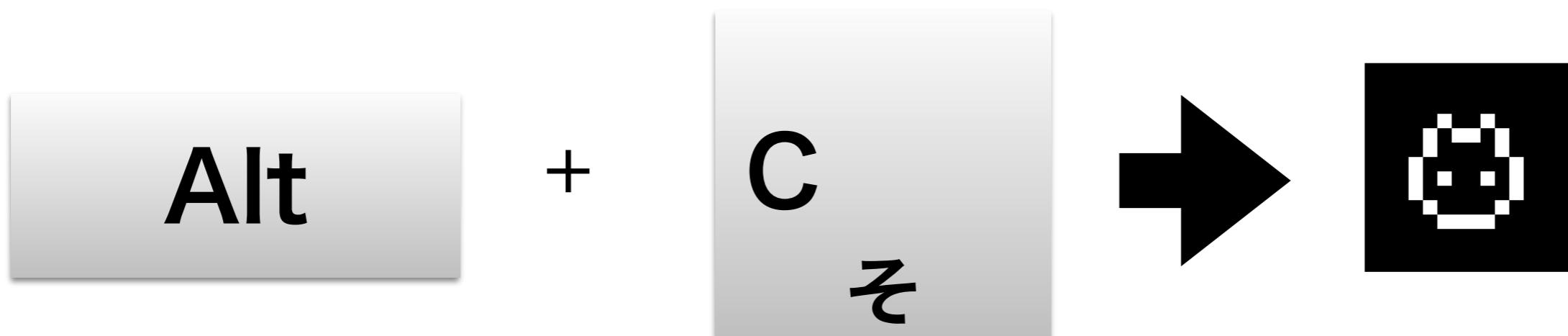


なにもしないよ

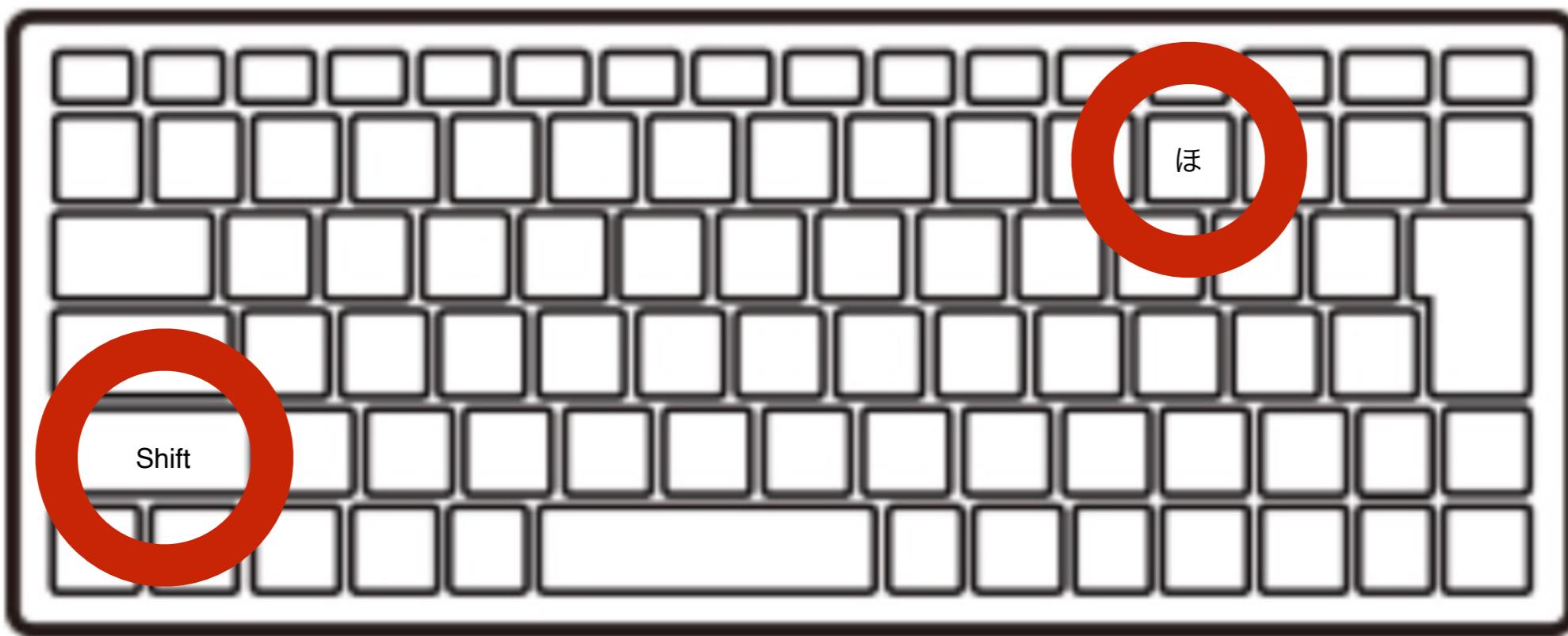
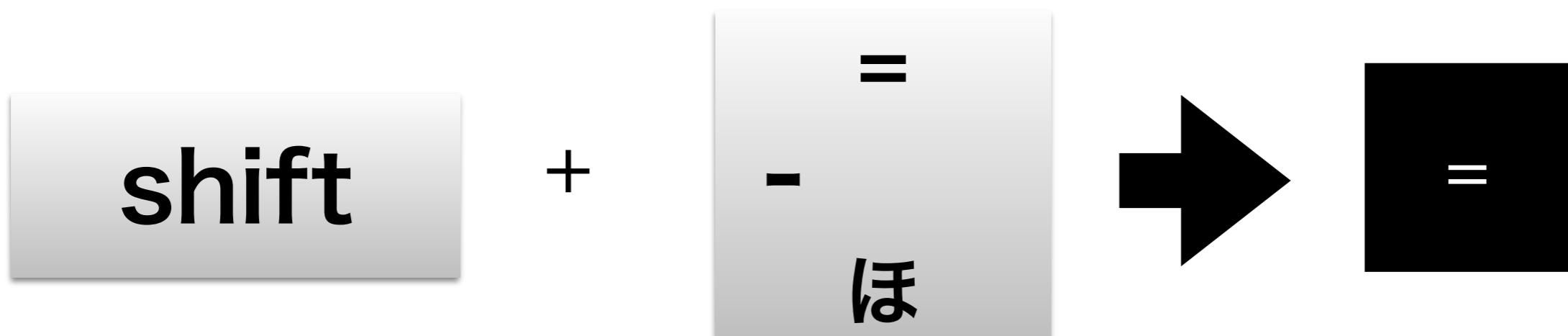


IchigoJam スペシャル

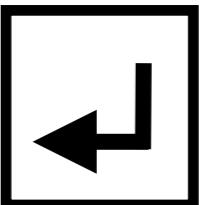
Alt (オルト) キーをおしながら「C」をおす



キーのうえにあるもじは
シフトキーをおしながらおす



ゲームづくり、はじめ！

10 CLS : X = 15 

↑ ↑
け Shift + ほ

なぜか10から

ラン（はしれ！／うごかす）

RUN

F5



はてなマークで“がめんにひょうじ

? × [←]
↑
Shift + め

なにができるかな？



コンピューターのきおく力

0か1か = 1bit

3万bit おぼえるよ (0-255の数で4千コ)



へんすう :

かず

T

1

4

1

2

1

7

かず

9

9

5

9

15

9

5

コンピューターのRAM（ラム）

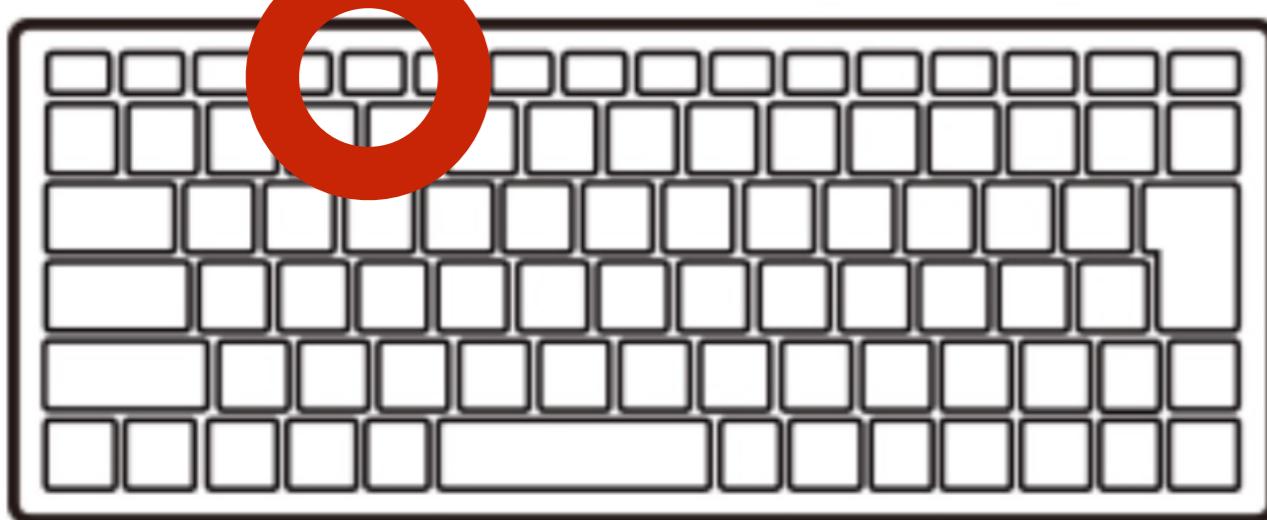
	IchigoJam	ノートPC
RAM (byte)	4KB	4GB
RAM (bit)	32,768bit (3万)	34,359,738,368bit (340億)

ノートPCは、 IchigoJamの100万倍の記憶力

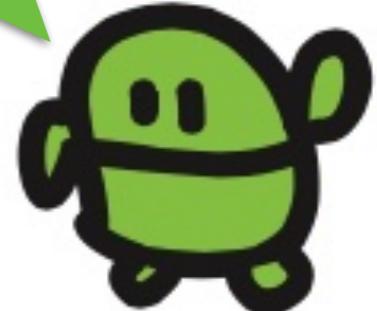
リスト（プログラムみせて）

LIST

F4



おもいだしてるよ



タ"ブル クオート
Shift+2

コンマ
ね

↓

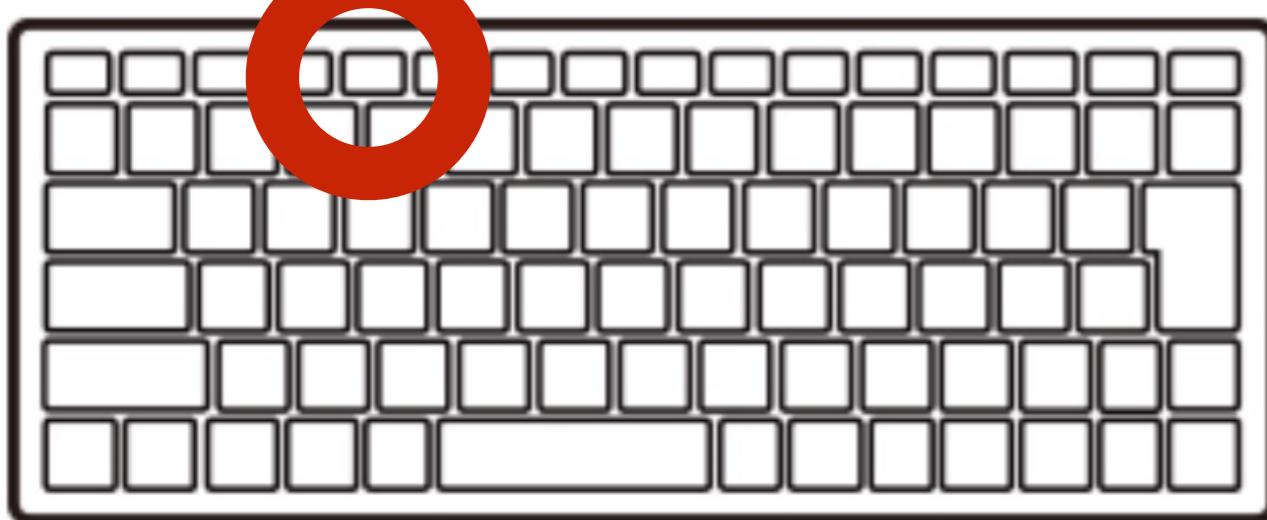
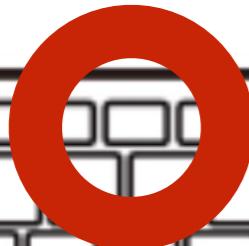
2 0 L C X , 5 : ? " 0 " ←

↑ ↑

Shift+め Alt+C

ハテナ

F5



じぶんキャラ

かっこ

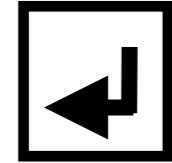
Shift+8

かっこ

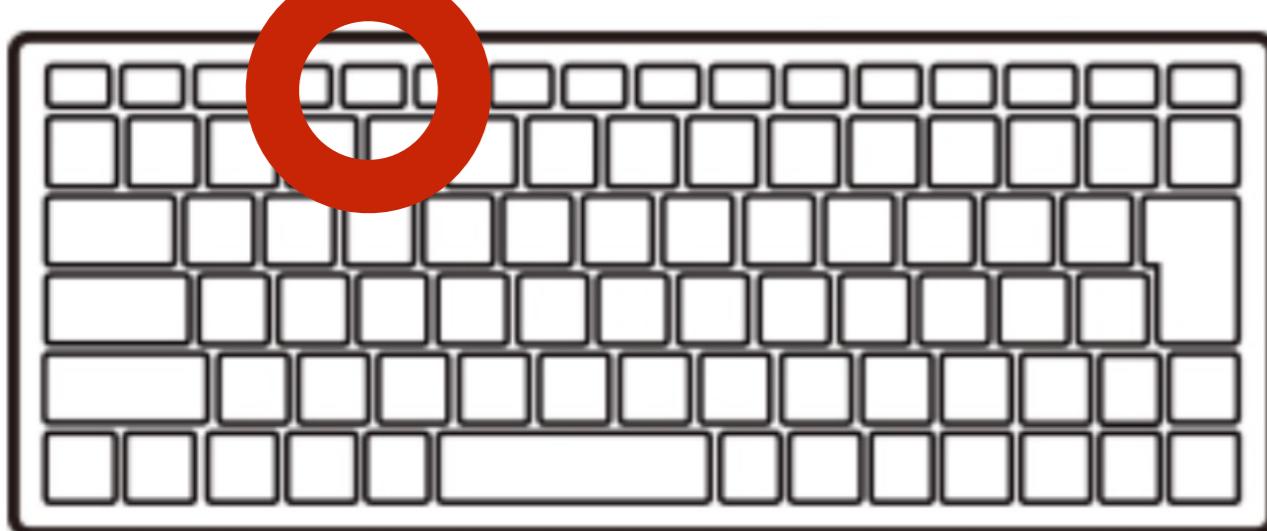
Shift+9

ダブルクオート

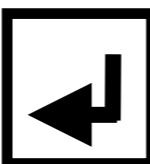
Shift+2

30 LC RND(32),23:"?"*"
↓ ↓ ↓
30 LC RND(32),23:"?"*"
↓ ↑ ↑ 
↓ ↑ ↑
ね Shift+め Shift+け
コンマ ハテナ アスタリスク

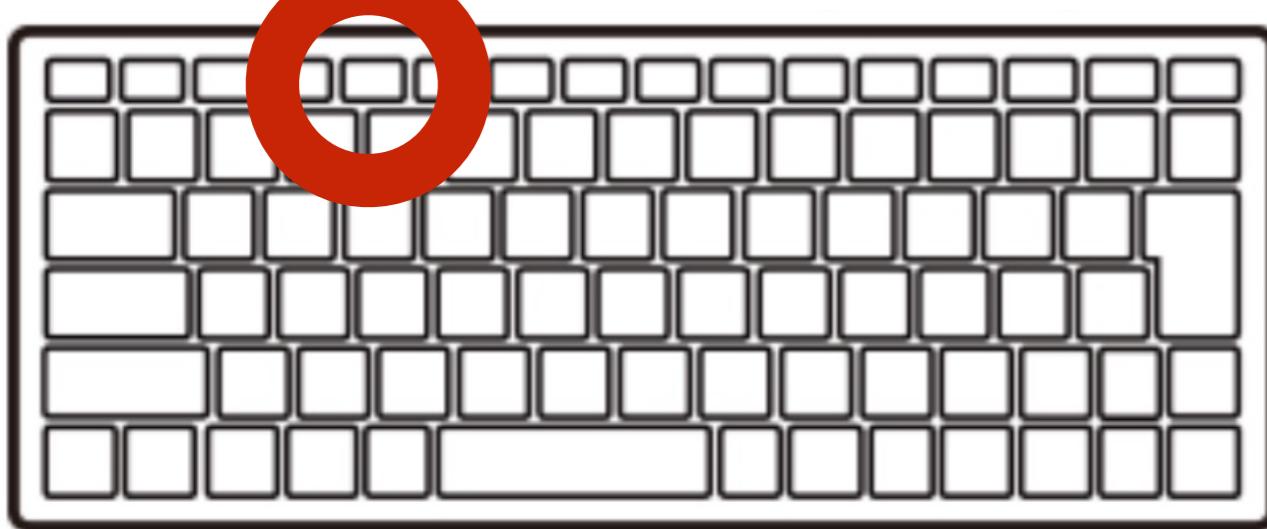
F5 なんとかおす



てきキャラ

40 GOTO 20 

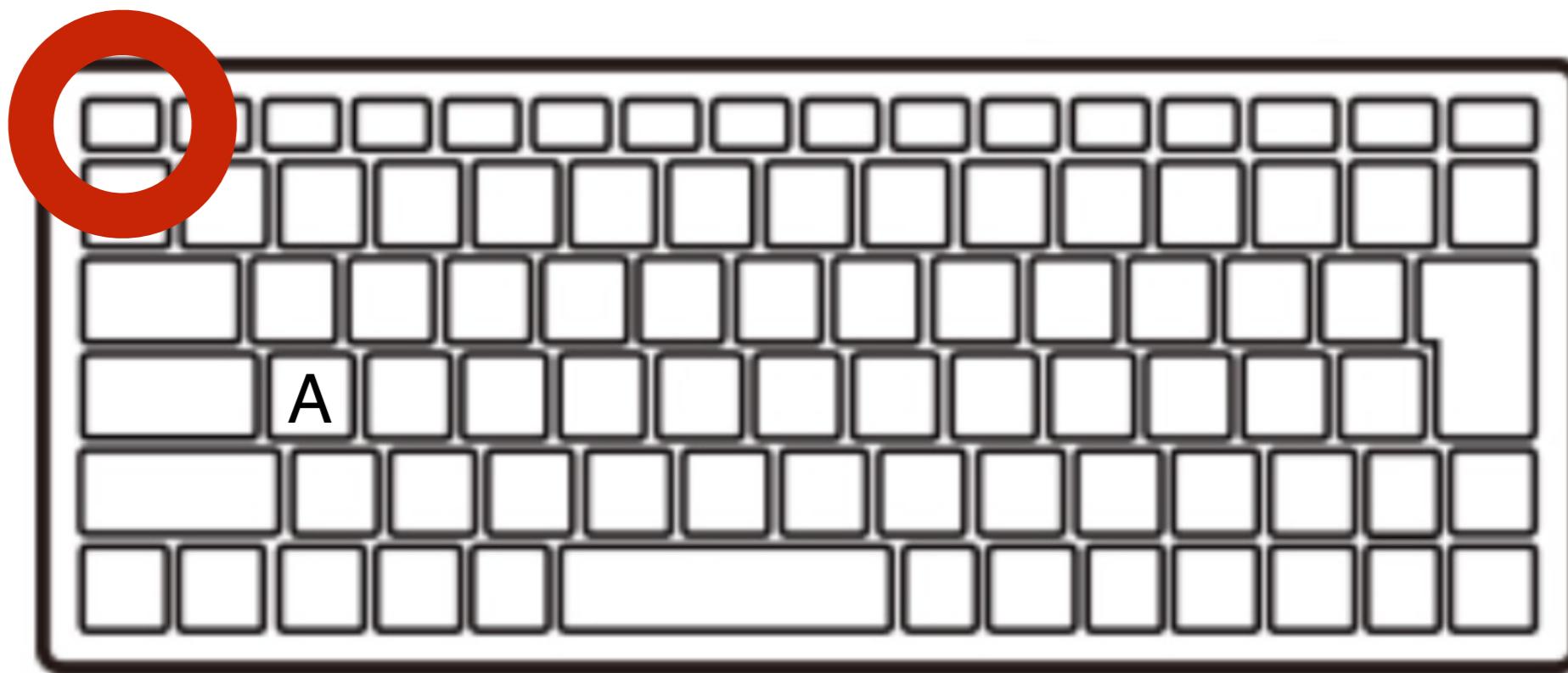
F5



! ?

とまって！エスケープキー

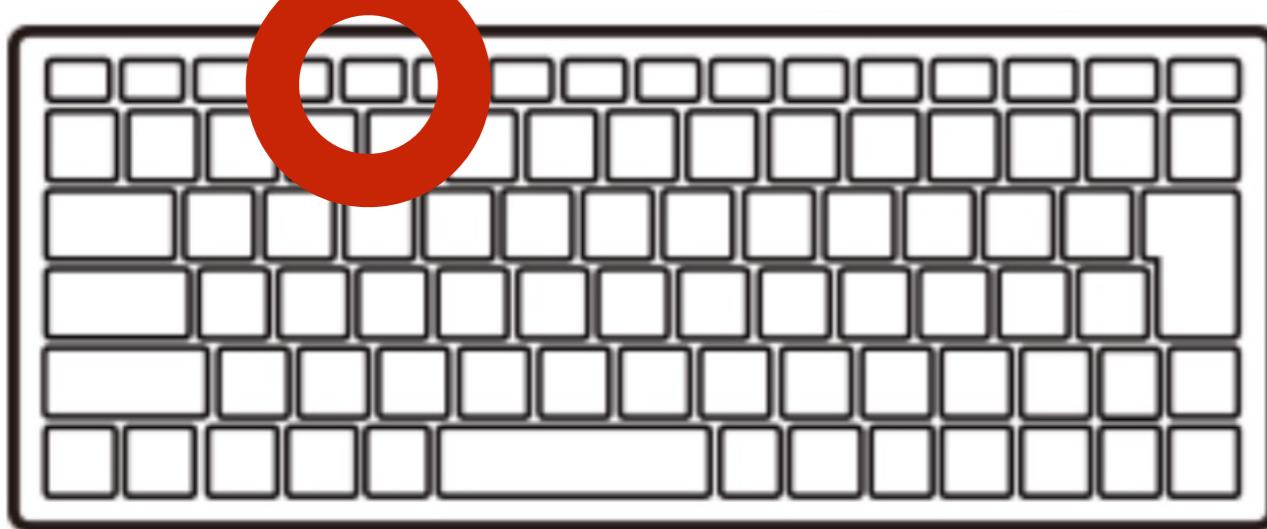
[ESC] + -



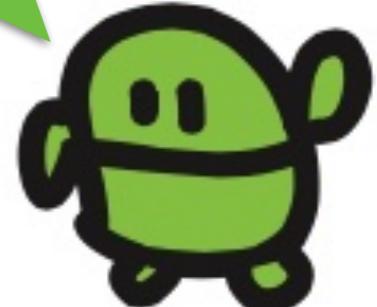
リスト（プログラムみせて）

LIST

F4

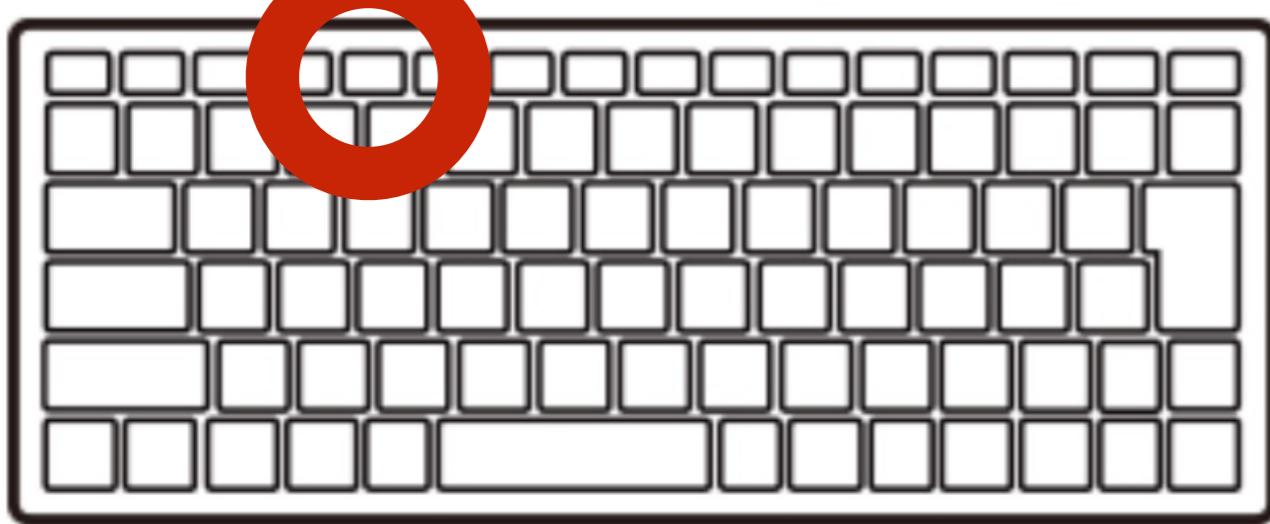


はやすぎた？



35 WAIT 3 ↵

F5



スピードちょうどいい

イコール

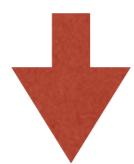
かっこ

かっこ

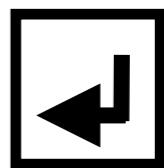
Shift+ほ

Shift+8

Shift+9



36 $X = X - \text{BTN}(28) + \text{BTN}(29)$



ほ

マイナス



れ

Shift+プラス

とめる (ESC)

みる (F4)

うごかす (F5)

カーソルでそまさ

かっこ

Shift+8



かっこ

Shift+9



39 IF SCR(X,5) END ←

とめる (ESC)

みる (F4)

うごかす (F5)

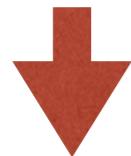
あたりはんてい

チート？ バグをなおそう



イコール

Shift + ほ



37 $x = x \& 31$ ↵



Shift + 6

アンド

エンター、F5

バグつぶし

ケ" - ム でき た !



プログラムのつくりをかくにん

さいしょだけ

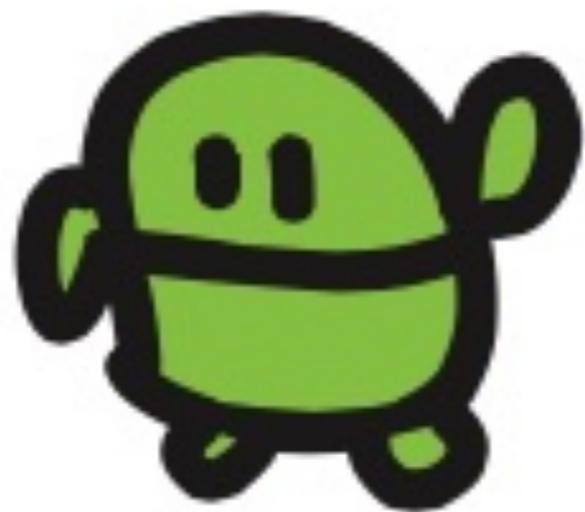
LC RND(32),23:?"*"
WAIT 3
X=X-BTN(28)+BTN(29)
IF SCR(X,5) END
GOTO 20
A blue vertical bar on the left has green arrows at the top and bottom. A blue horizontal bar surrounds the last four lines of code. A red arrow points from the 'END' keyword to the right. A blue arrow points from the '0' in 'GOTO 20' to the '0' in 'CLS : X=15'. A black arrow points from the '0' in 'CLS : X=15' to the word 'さいしょだけ'."/>

```
10 CLS : X=15
20 LC X,5;"0"
30 LC RND(32),23:?"*"
40 WAIT 3
50 X=X-BTN(28)+BTN(29)
60 IF SCR(X,5) END
70 GOTO 20
```

じぶんキャラのいちに、なにかあれば、おわる (END)

アプリのきほん！

ケ" - ムたいかい !



```
10 CLS : X=15
20 LC X,5;"@"
30 LC RND(32),23:"?"  
40 WA IT 3
50 X=X-BTN(28)+BTN(29)
60 X=X&31
70 IF SCR(X,5) END
80 GOTO 20
```

F4でひょうじ
かえたら、エンター

F5

なんいどアップ

```
10 CLS : X=15
20 LC X,5;"@"
30 LC RND(32),23;"♪♪♪"
40 WA IT 6 
50 XX=X-BTN(28)+BTN(29)
60 X=X&31
70 IF SCR(X,5) END
80 GOTO 20
```

F4でひょうじ
かえたら、エンター

F5

なんいどダウン

```
10 CLT : CLS : X=15
20 LC X,5:?""
30 LC RND(32),23:?""
40 WAIT 10-TICK() / 120
50 X=X-BTN(28)+BTN(29)
60 X=X&31
70 IF SCR(X,5) END
80 GOTO 20
```

F4でひょうじ
かえたら、エンター

F5

だんだんはやく

IoTたいけん

かわくだりをネットせつぞく



IoTの「I」

```
LIST 39  
39 IF SCR(X,5) IOT.OUT TICK():END  
RUN
```



IOT.OUT 10000

エンター、F5

チート

IoTの「I」

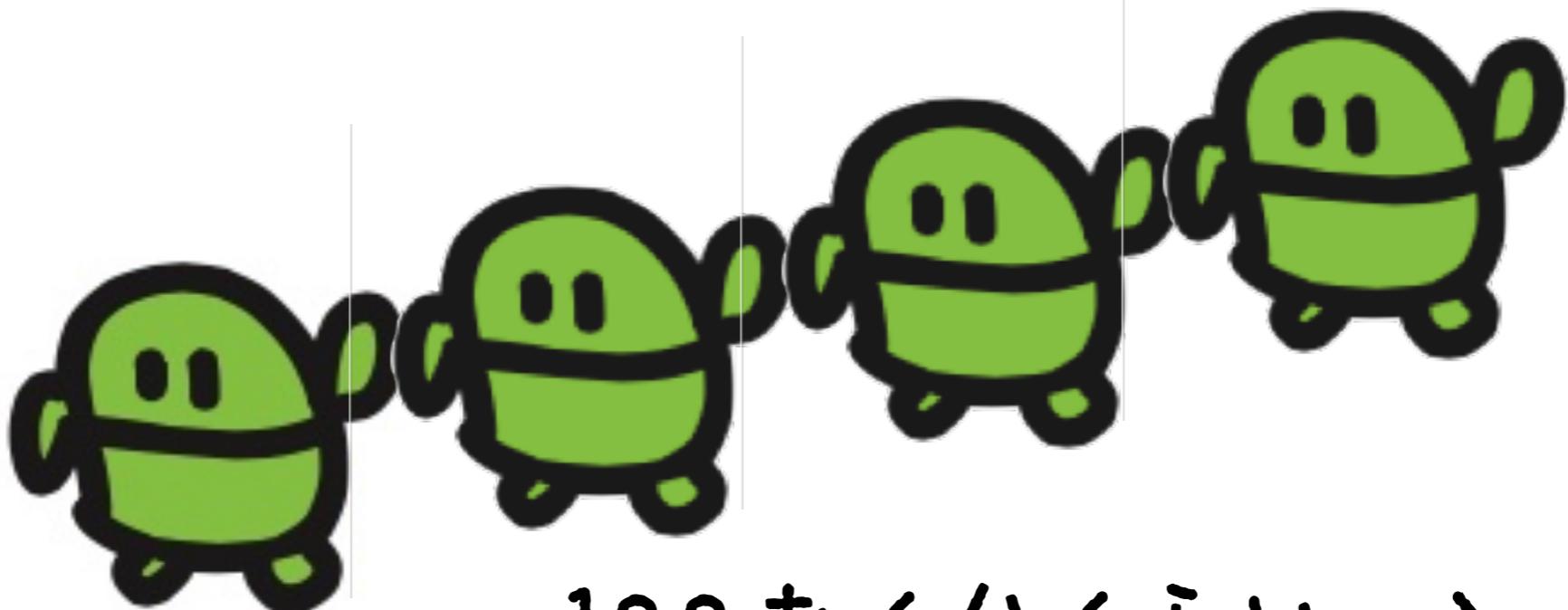
```
LIST 36  
36 X=X-BTN(28)+BTN(29)+IOT.IN()  
RUN
```



<https://c4.sabae.cc/sakura/>

スマホからも
うごく！？

インターネットは
コンピュータが
たくさんつながったもの



100 おく台くらい -> 1 ちよう台

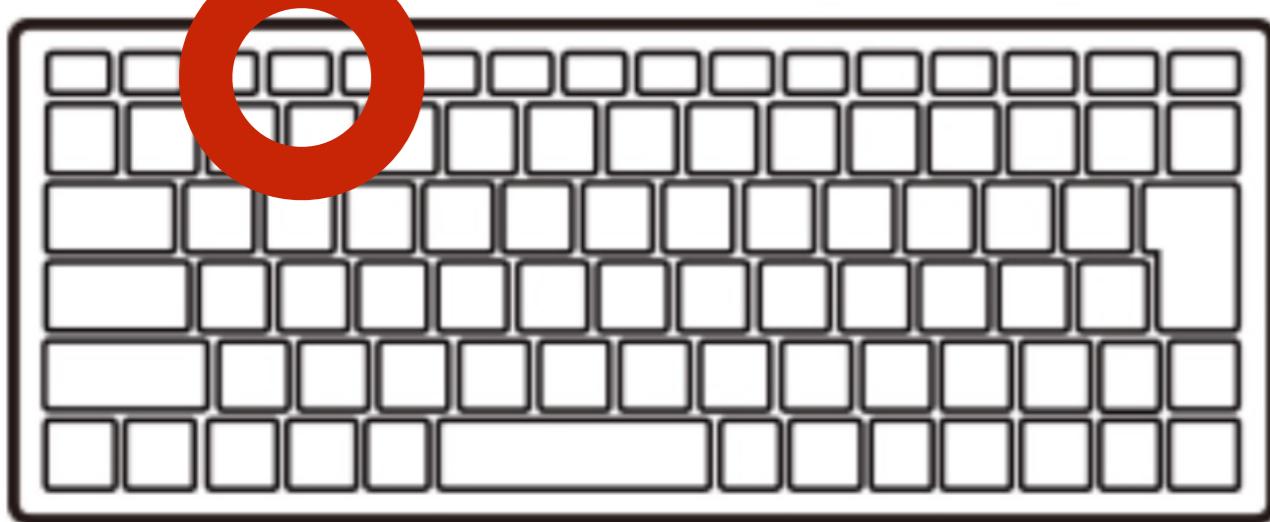
いちばんちいさな
ネットをつくろう



ほぞん（プログラム書き込み）

SAVE1 ↵

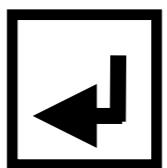
F3



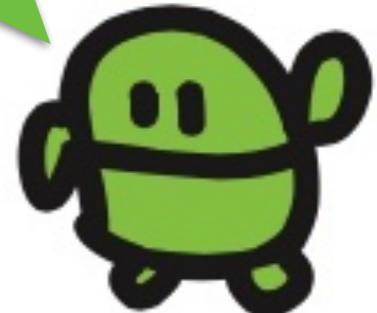
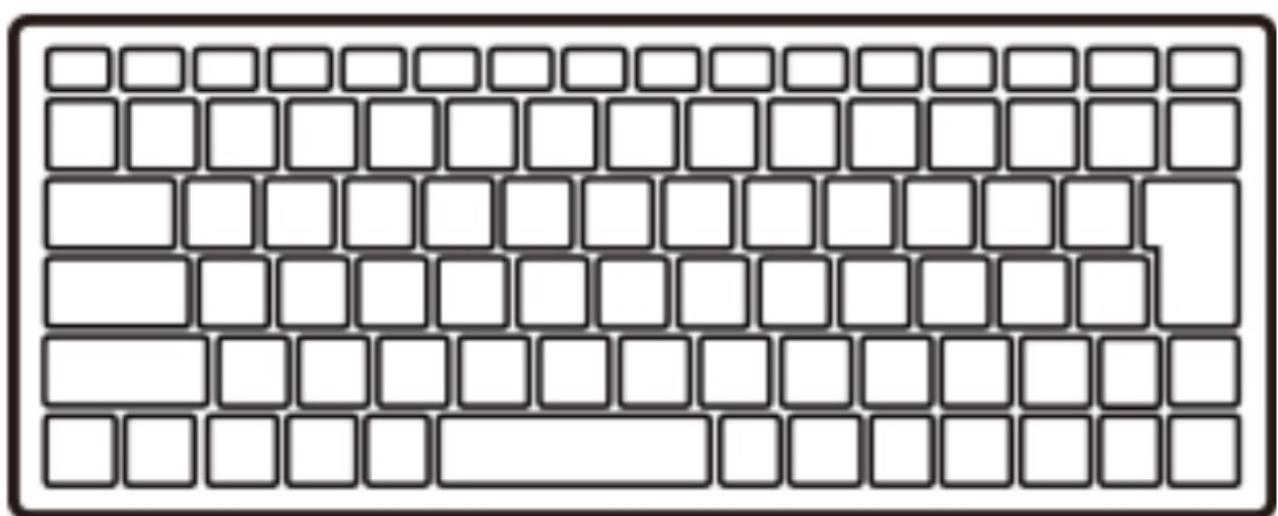
F3、1、エンター



さいしょから

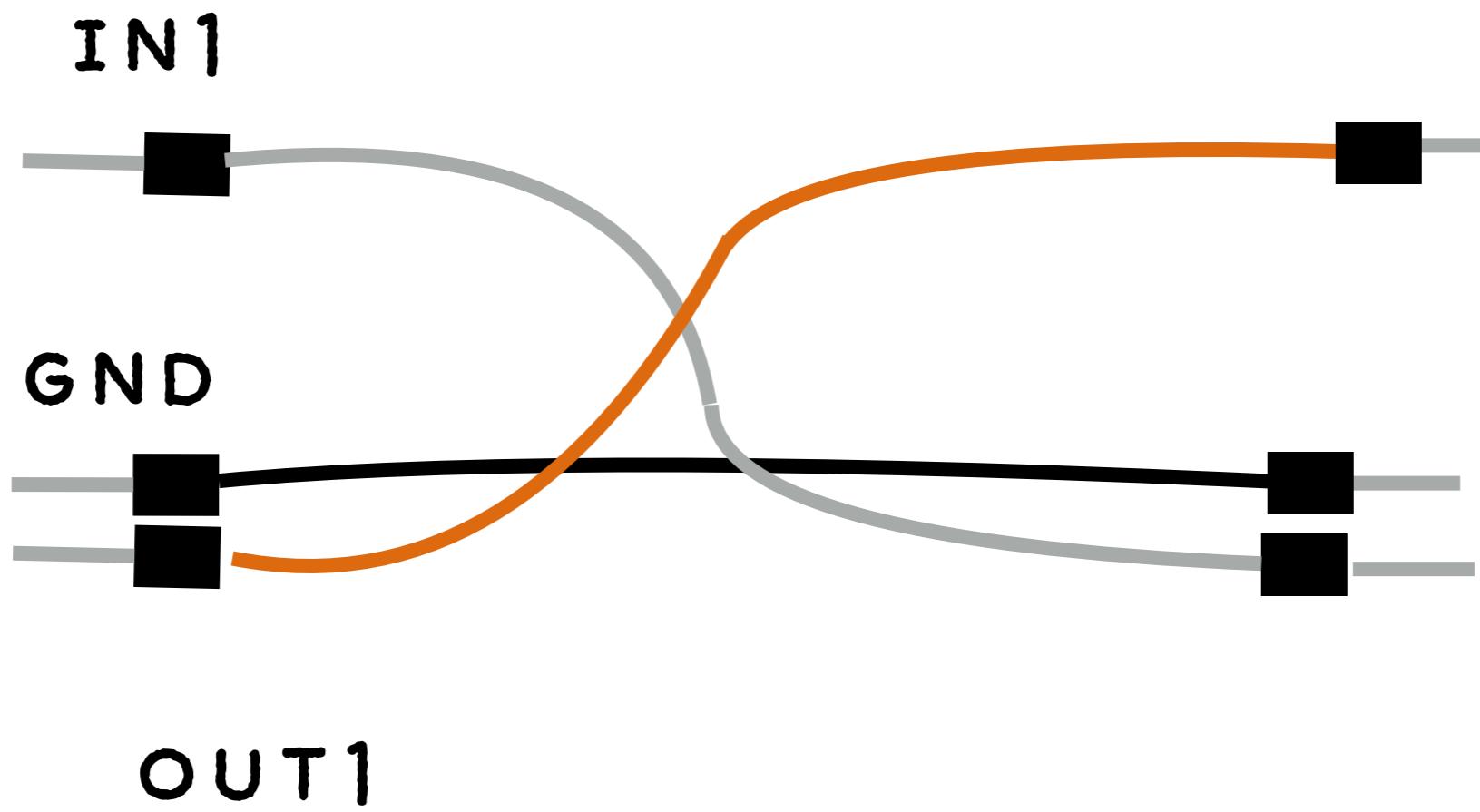
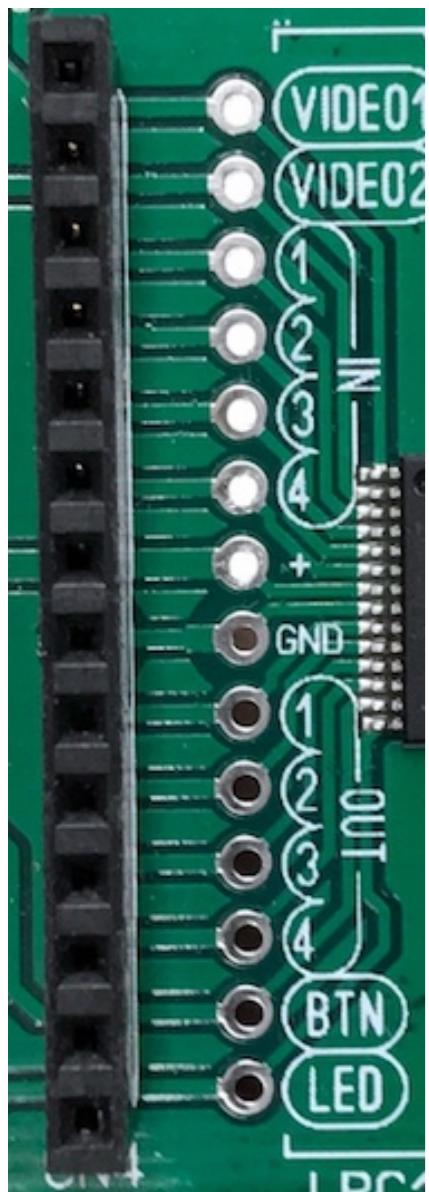
NEW 

ほぞんしたのは
きえないよ



Aさんジヤン1Pーせん、3つとりだそう

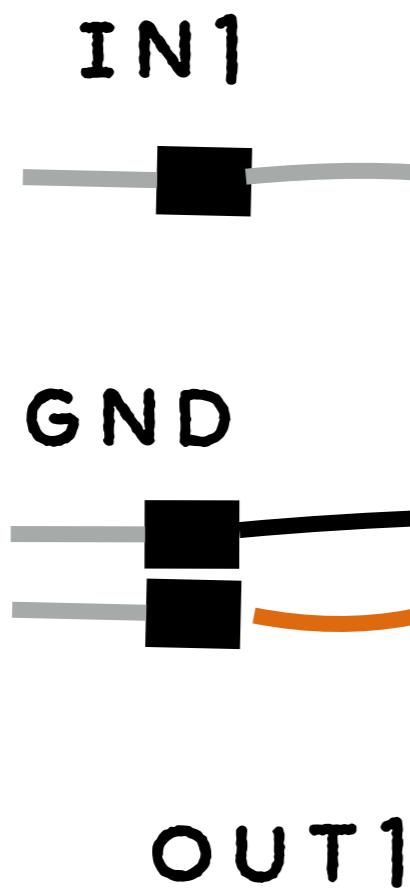
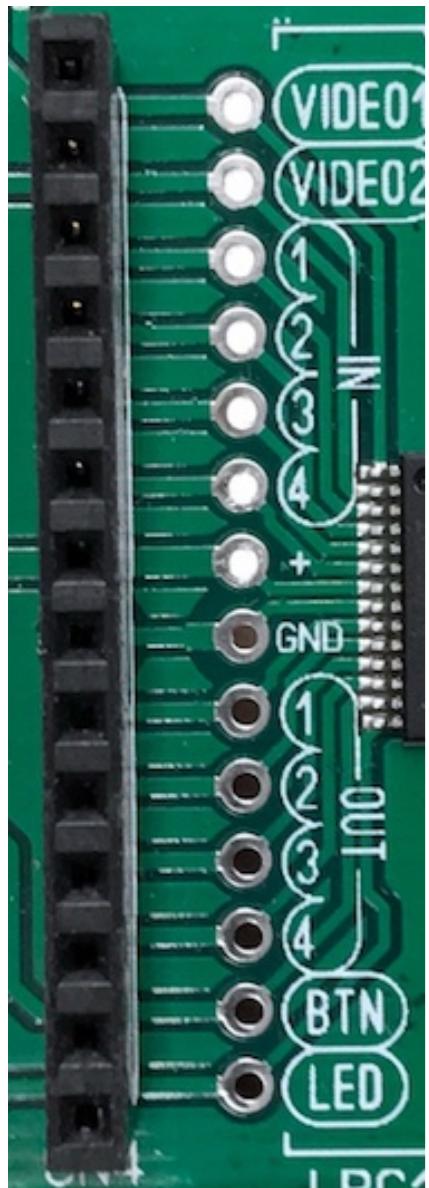
IchigoJam A



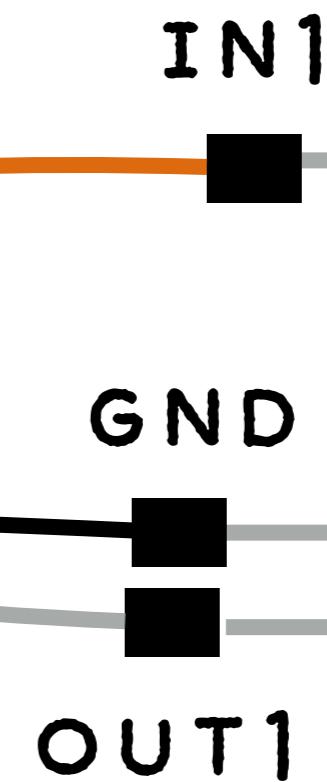
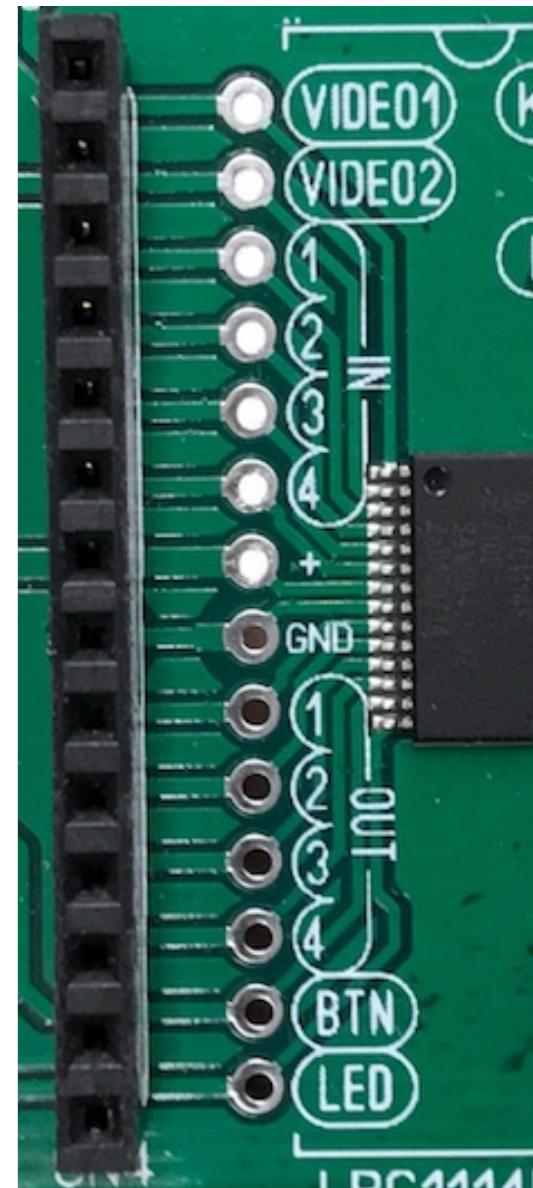
CN4

Bさんジャンパーせんをつなごう

IchigoJam A



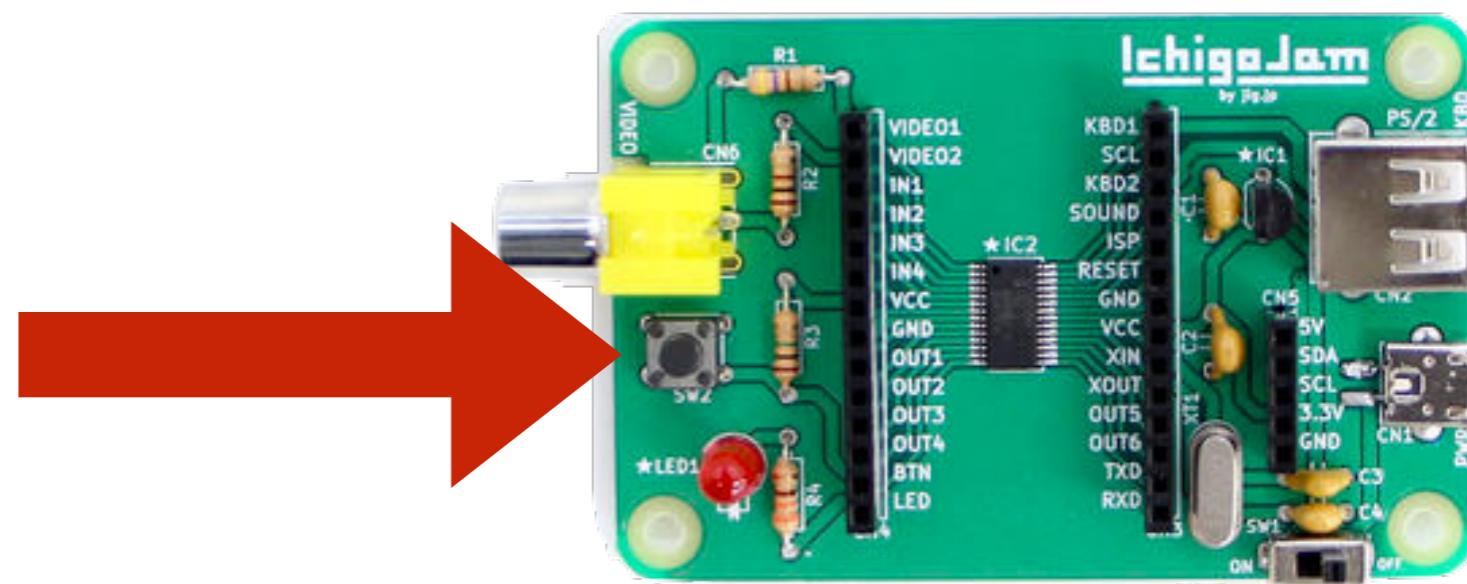
IchigoJam B



CN4

```
1 ?IN(1) : OUT1, BTN()
2 WAIT10 : GOT01
```

1 ?IN(1):OUT1,BTN():CONT [←]
RUN



とめる (ESC)

みる (F4)

うごかす (F5)

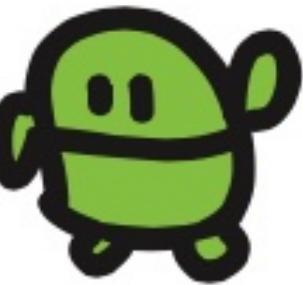
ボタンで
かいわしよう

0と1だけで"つたわる？



オリジナル
IoTデバイスをつくろう





IoTとは

ネットにつながったロボット

しげき
(INPUT)

コンピューター

アクション
(OUTPUT)

暑さ (温度センサー)

うで (サーボ)

感触 (ボタン)

プログラミング

かお (画面)

明るさ (光センサー)

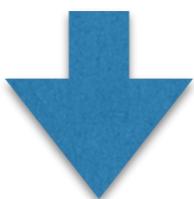
こえ (音)

ネット

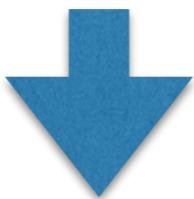
ネット

IoTのはつめいの仕方

こまつた



こんなのがあるといいかな？



つくってみよう



IoT のサンプル



ねこ健康、ネコトイレIoT



→ ねこがトイレにいくと通知、統計とれば病気予測も！

<http://fukuno.jig.jp/1518>

6:38

“アイデアを形に”
鯖江発 小型コンピューター



見回りいらず
イノシシIoT



NHK
おはよう日本
(東海北陸地区)
2015.12.7

IoT × 火災報知器 by 創電

住宅用火災警報器連動の火災通報システム

万が一発生する火災に対し、現場にいない場合でも火災発生を素早く把握することができ、近隣住民、関係者へいち早く通報することが可能となり、被害の拡大および2次災害を防ぐことを目的としたシステムです。



無線通信端末機

火守くん
HOMORI-KUN



火守くん SO-DEN

サイズ：幅160×高さ80×奥行き35(mm) / 重さ：250g / カラー：ブラック・ホワイト
<別>パナソニック電工(株)
<特許出願中>

SAKURA internet

導入事例・構成例

> 導入事例から探す > 構成例から探す

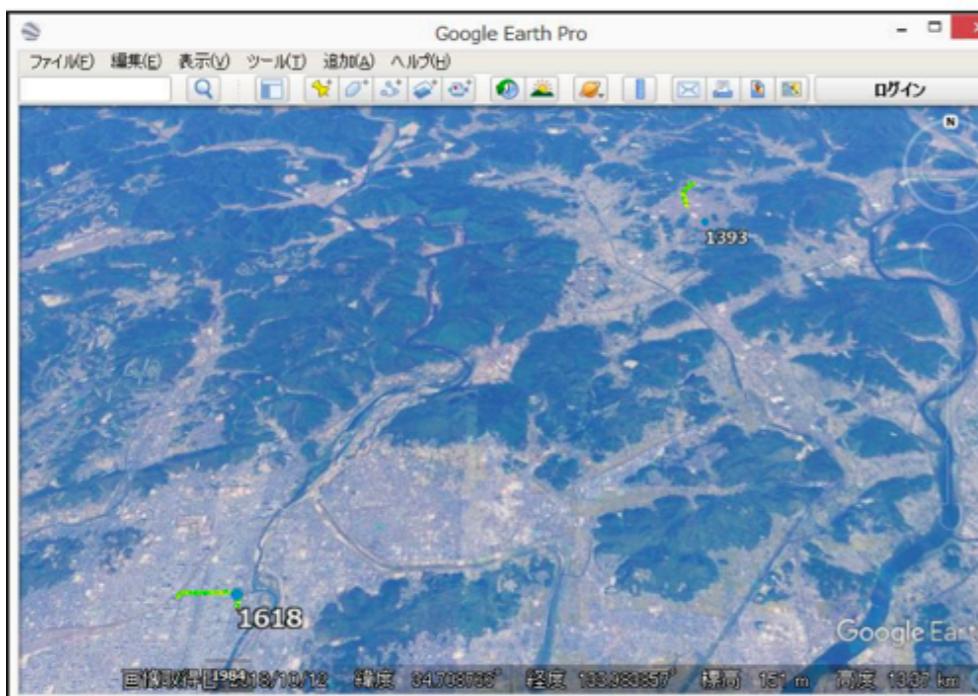
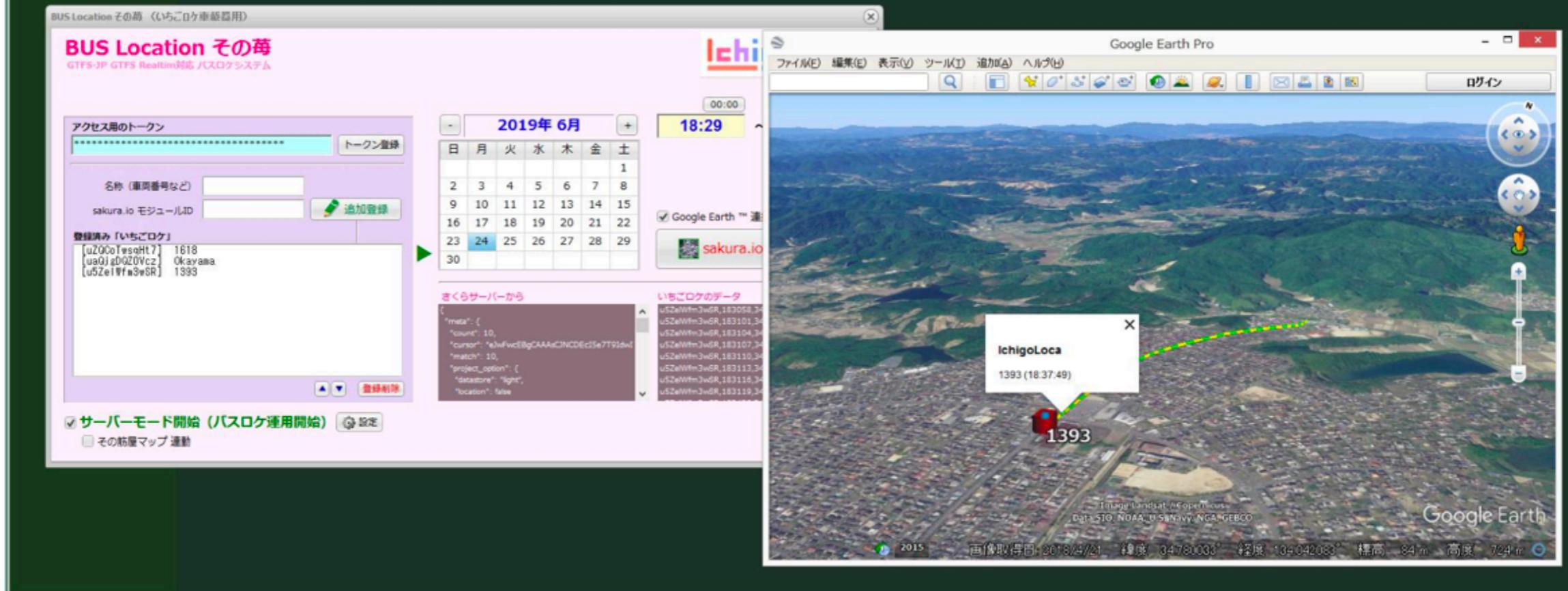
✉ サービスのご利用に関する
ご相談・お問い合わせはこち



Hana道場生まれの
オープンイノベーション

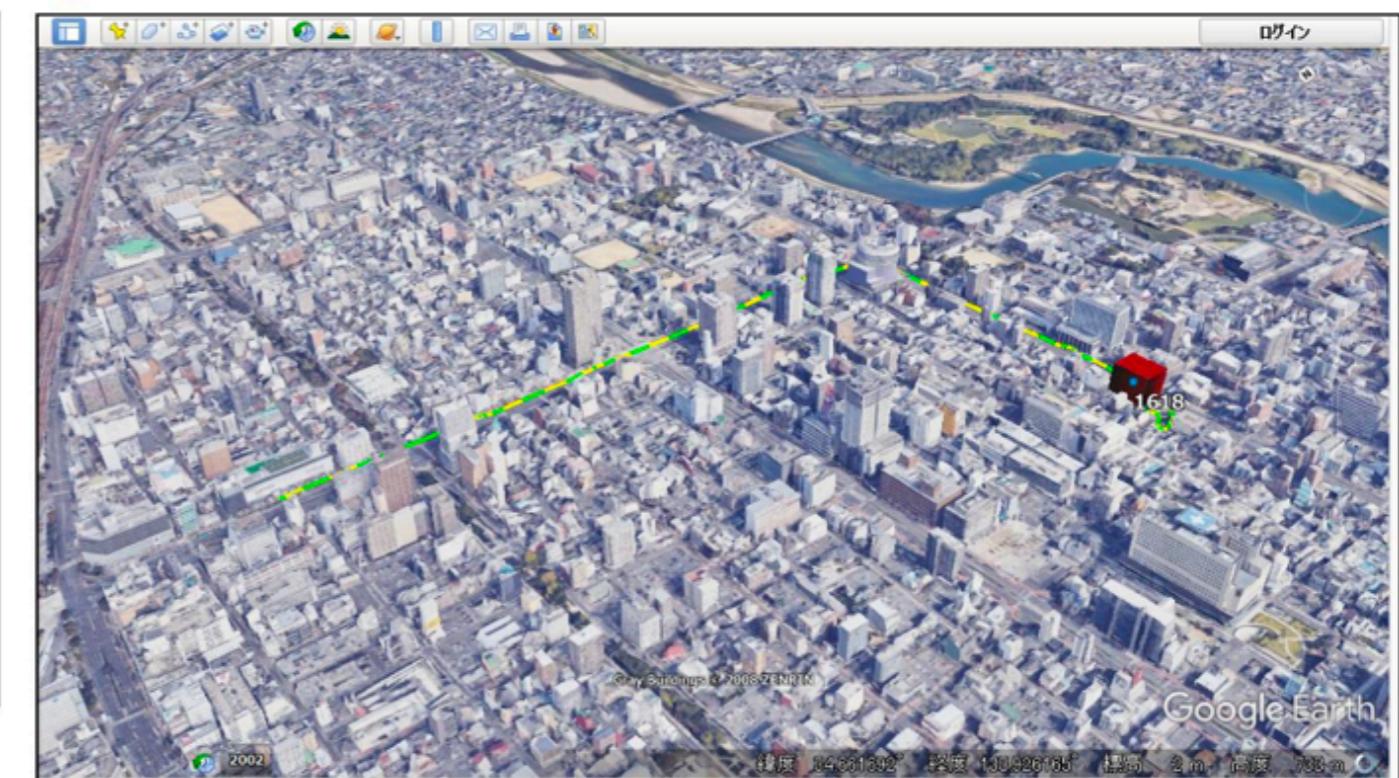
BUS Location その苺

GTFS-JP GTFS Realtime相棒 バスロケーションシステム



オープン・バスロケ車載器

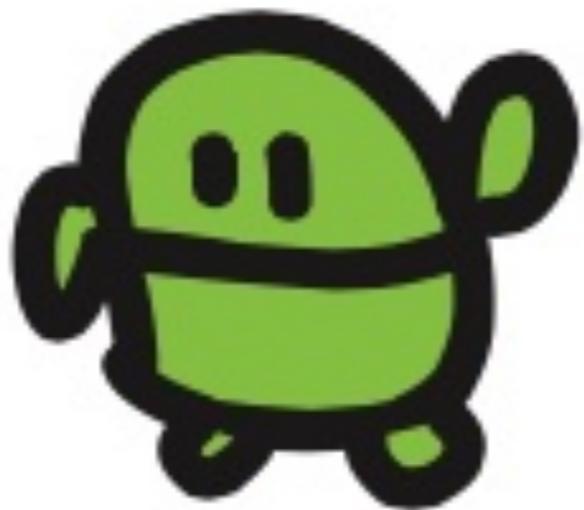
いちごロケ



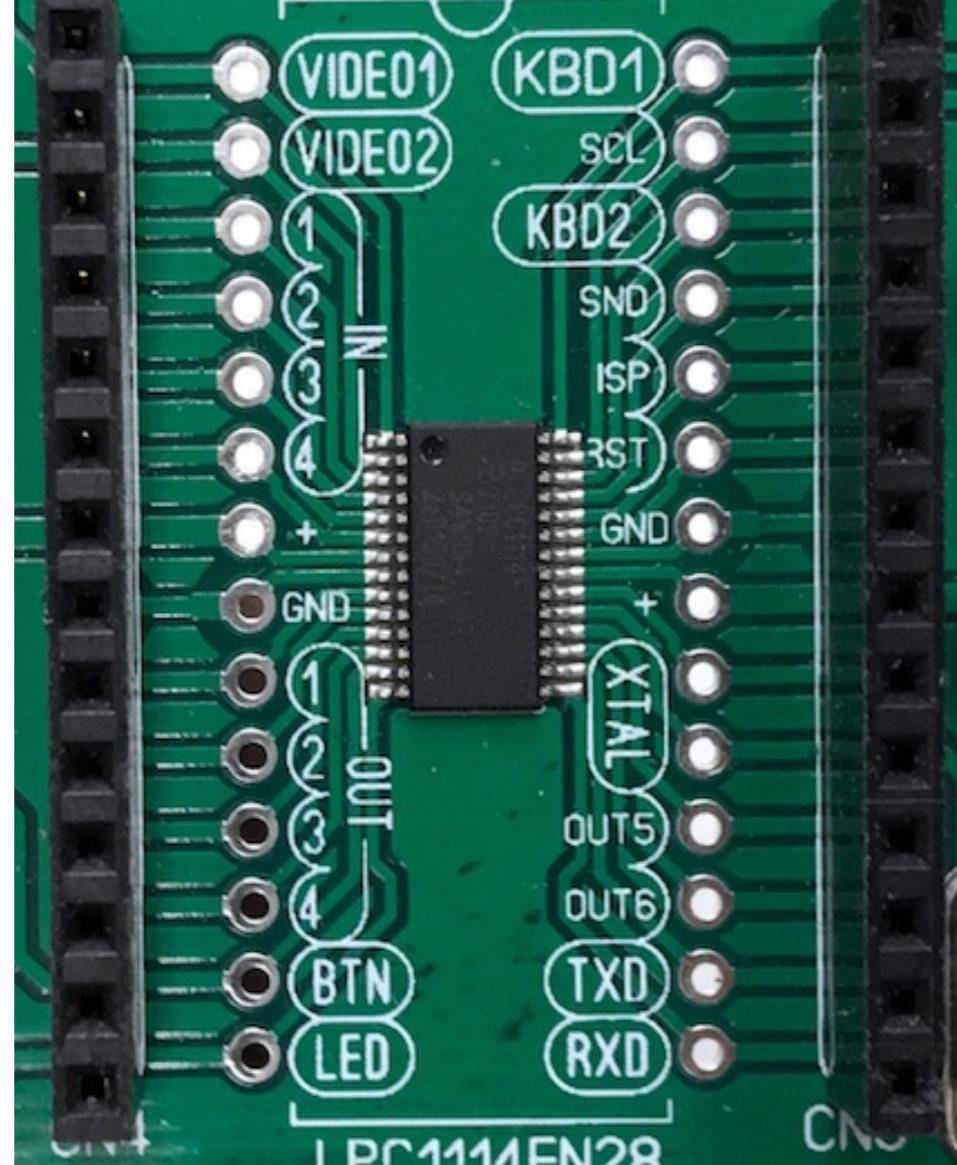
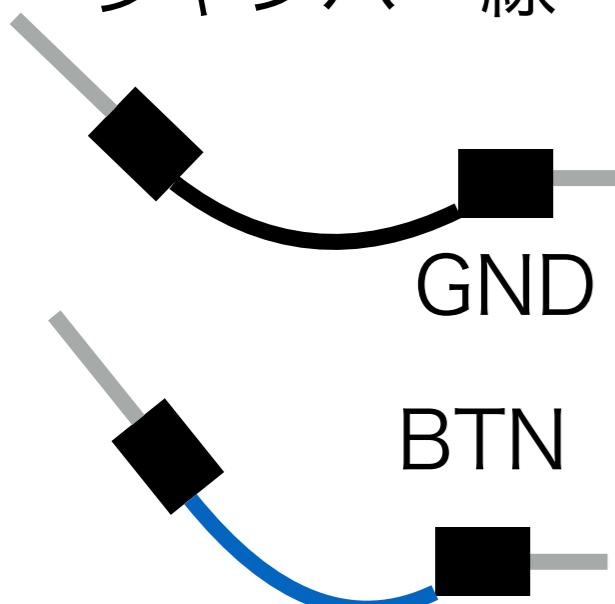
バス x IoT x オープンデータ

乗換案内が遅れも加味した乗換案内の実現 Google連携

プログラムサンプル



CN4の
一番下と
下から7番目に
ジャンパー線



せっしょくセンサー



<https://sabae.cc/sakura/>

CN4

```
1 IF BTN() ?1 : IOT.OUT1 : WAIT120
2 GOTO1
```

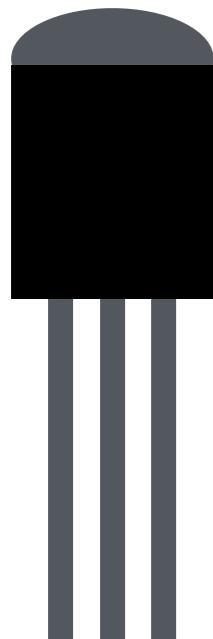
```
1 IF BTN() = 0 CONT
2 IOT.OUT 1
3 IF BTN() CONT
4 GOTO 1
```

ふたつの先端を接触させると
ネットに1を送信

おんどセンサーをつなごう

MCP9700-A/TO

30円



たいらなほうを上



いちばん下、すこしまげる

(OUT8)
IN1
IN2
IN4
(OUT11)

CN4



NEW

10 OUT8,0:OUT11,1

20 A=ANA(2):?A

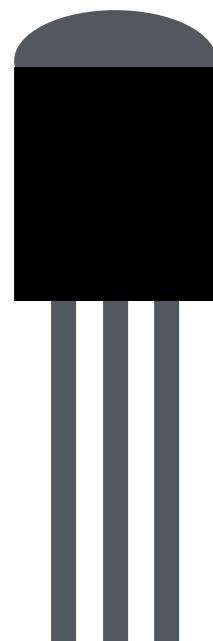
30 GOT020

RUN

あついところチェック

MCP9700-A/TO

30円



たいらなほうを上



CN4



10 OUT8,0:OUT11,1

20 A=ANA(2):?A

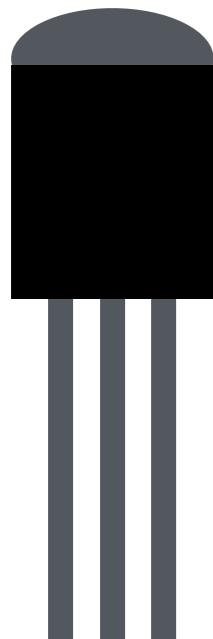
30 WAIT30:GOTO20

25 LED A>190

IoT おんどセンサー

MCP9700-A/TO

30円



たいらなほうを上



いちばん下、すこしまげる

CN4



10 OUT8,0:OUT11,1

20 A=ANA(2):?A

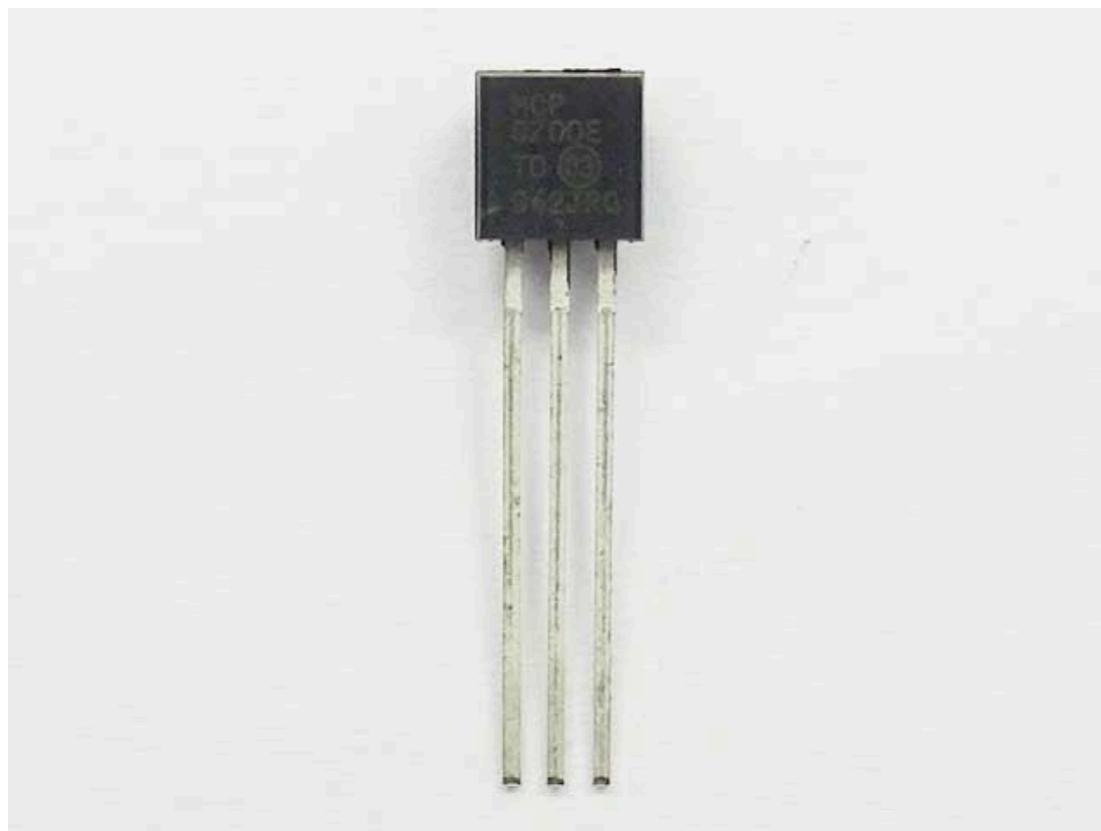
30 WAIT30:GOTO20

26 IOT.OUT A



温度センサー（精度が違う）

40円～500円



-40～125°C **$\pm 4^\circ\text{C}$**

40円



-40～105°C **$\pm 0.5^\circ\text{C}$**

500円

スマホからコントロール

```
1 N=IOT.IN()
2 IF N?N:LED N-1:BEEP N
3 GOTO 1
```



<https://sabae.cc/sakura/>

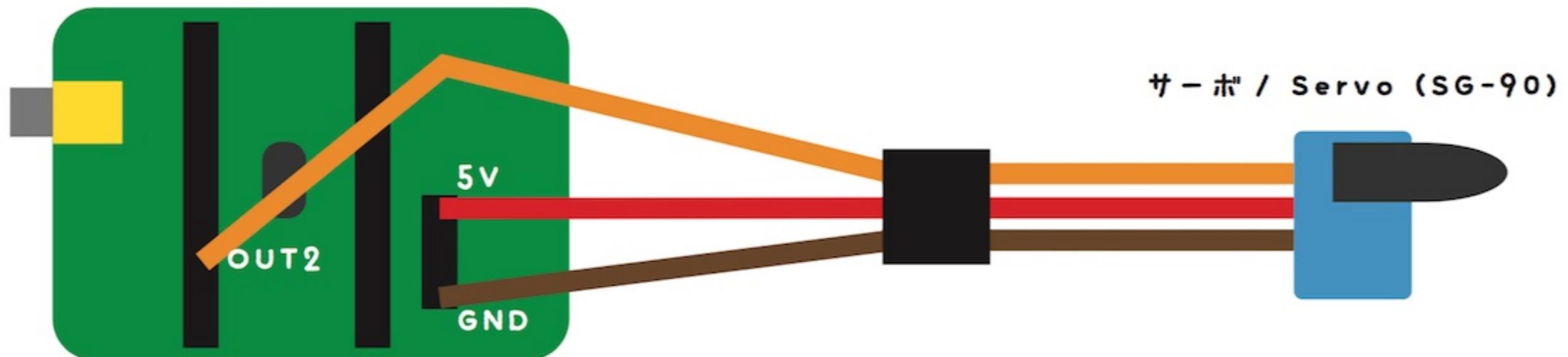
自分の番号を選択

2を送ると点灯

1を送ると消える

ネットからの受信

サー ボをうごかす



1. オレンジは OUT2 へ
CN4 したから 5 ばんめ
Orange-OUT2(CN4)
2. あかは 5V へ
CN5 いちばんうえ
Red-5V(CN5)
3. ちゃいろは GND へ
CN5 いちばんした
Brown-GND(CN5)

NEW

```
10  PWM 2,140 : WAIT30
20  PWM 2,70 : WAIT30
30  GOT010
RUN
```

ボタンでサーキ

```
10  PWM 2,140:WAIT30
20  PWM 2,70:WAIT30
25 IF BTN()=0 CONT
30 GOTO10
```

エンター

F5でためそう

ボタンを押すと？

スマホ・ネットからサーボ

```
10  PWM 2,140:WAIT30
20  PWM 2,70:WAIT30
25  IF IOT.IN()=0 CONT
30  GOTO10
```

エンター

F5でためそう



スマホ・ネットからサーボ

```
1 N=INPUT,IN()
2 IF N PWM2,N
3 GOTO 1
```

エンター

F5でためそう



BEEP

BEEP 5

BEEP 5,30

おとでる

BEEP (<おとのたかさ>), <おとのながさ>
<わしくは「IchigoJam BASIC リファレンス」



PLAY " C D E F G "

PLAY " *C D E 2 "

PLAY " C D E F G A B < C "

PLAY " T8000*C E G R "

PLAY " (MML)"

ミュージック・マクロ・ランゲージ (CDEFG = ドレミファソ)

くわしくは「IchigoJam BASIC リファレンス」

```
1  PLAY"CDE2CDE2GEDCDCD2"
2  IF SOUND()>0 CONT
3  IF IOT.IN()=0 CONT
4  GOTO1
```

SOUND() おとがなっているとき: 1

IOT.IN() データがきたら 0 いがい



IoTアラーム!?

スマホからコントロール

```
1 N=IOT. IN()
2 IF N=1 LED1
3 IF N=2 PLAY" CDE"
4 IF N=3 BEEP 30,30
5 GOTO 1
```



スマホから送る番号によって
やることをいろいろかえる

<https://sabae.cc/sakura/>

ネットからの受信

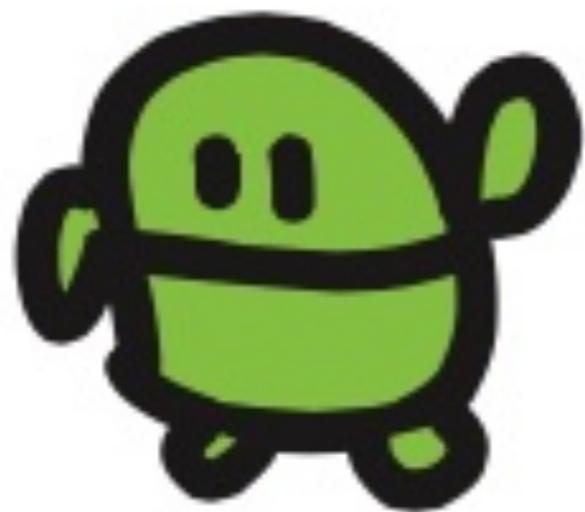
web アプリとつなごう

GitHub で **iothackbase** で検索！→Fork

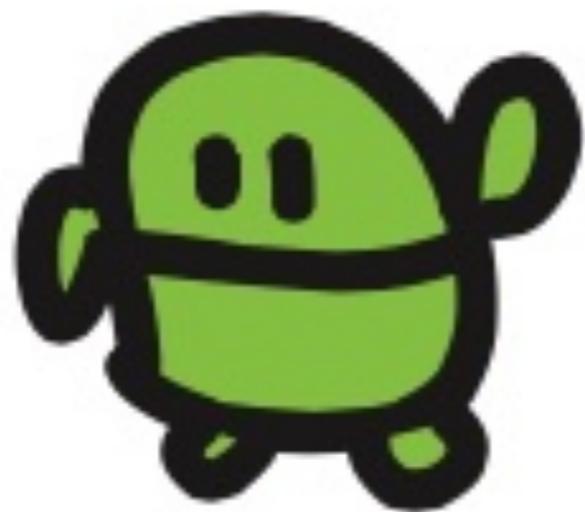
<https://github.com/taisukef/iothackbase/>

The screenshot shows a GitHub repository page for 'taisukef / iothackbase'. The URL in the browser bar is 'https://github.com/taisukef/iothackbase/'. The page title is 'taisukef / iothackbase'. The main navigation menu includes 'Pull requests', 'Issues', 'Marketplace', 'Explore', 'Watch' (0), 'Star' (0), and a circled 'Fork' button (1). Below the title, there are links for 'Code', 'Issues 0', 'Pull requests 0', 'Projects 0', 'Wiki', 'Security', 'Insights', and 'Settings'. The repository summary shows '5 commits', '1 branch', '0 releases', and '1 contributor'. At the bottom, there are buttons for 'Branch: master', 'New pull request', 'Create new file', 'Upload files', 'Find File', and 'Clone or download'. A commit history entry from 'taisukef' is shown, dated '35 seconds ago'. A red circle highlights the 'Fork' button.

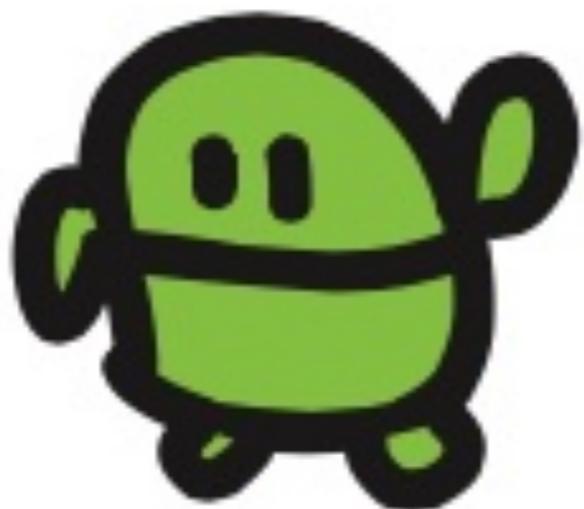
まとめ

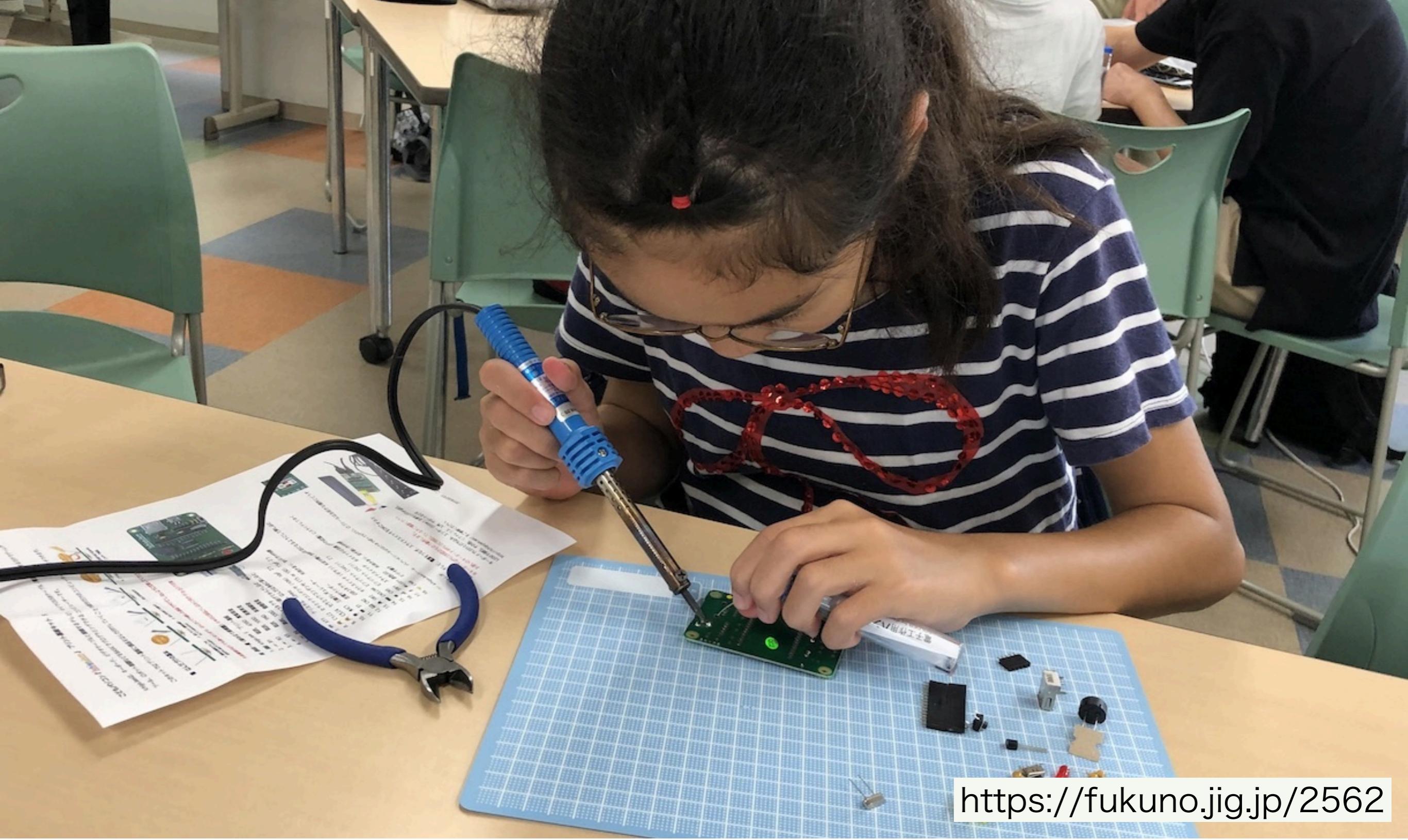


IoTは
じぶんでつくれる！



パソコンも
じぶんでつくれる



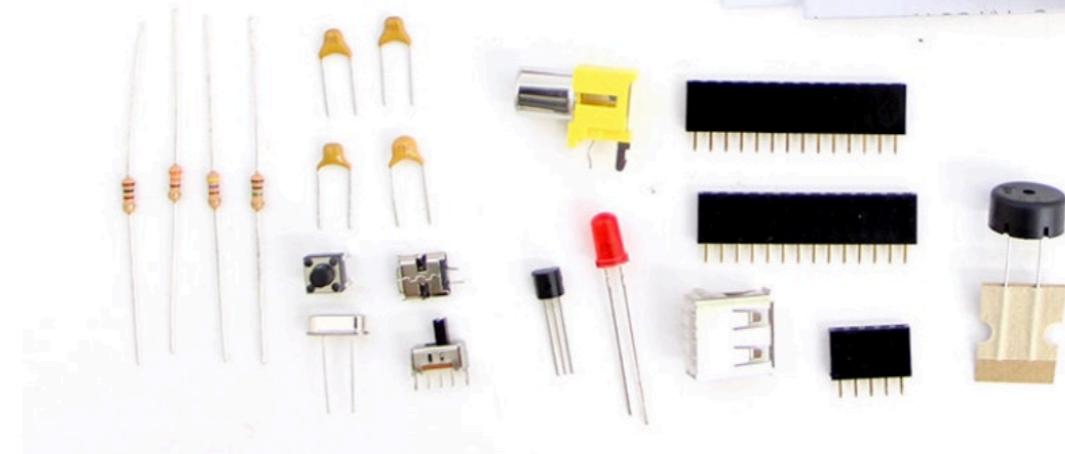
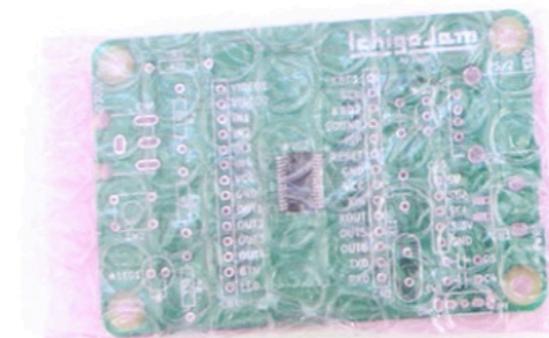


<https://fukuno.jig.jp/2562>

じぶんでつくるう！

じぶんのパソコン！

IchigoJam

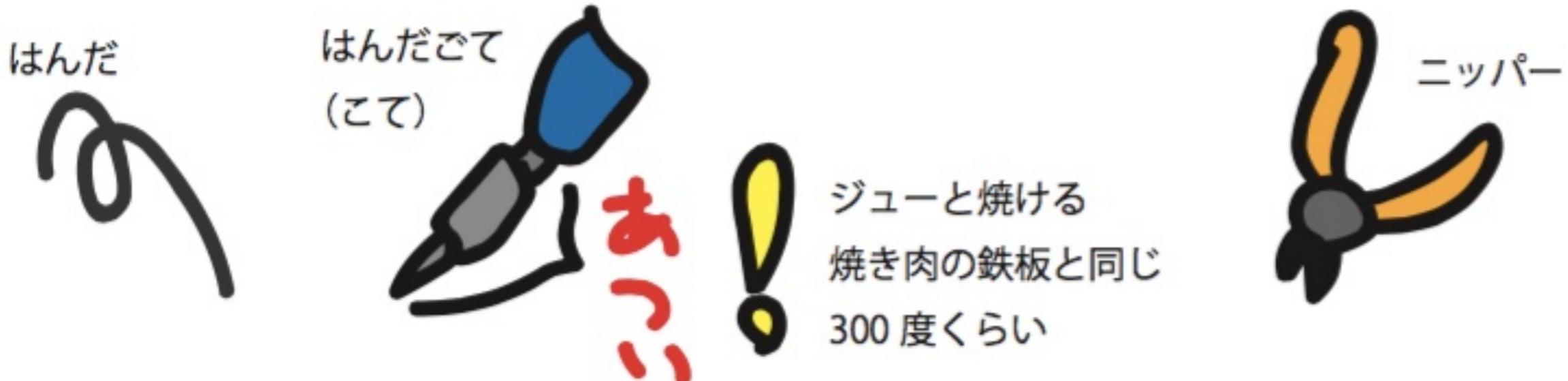


じぶんでくみたてるパソコン

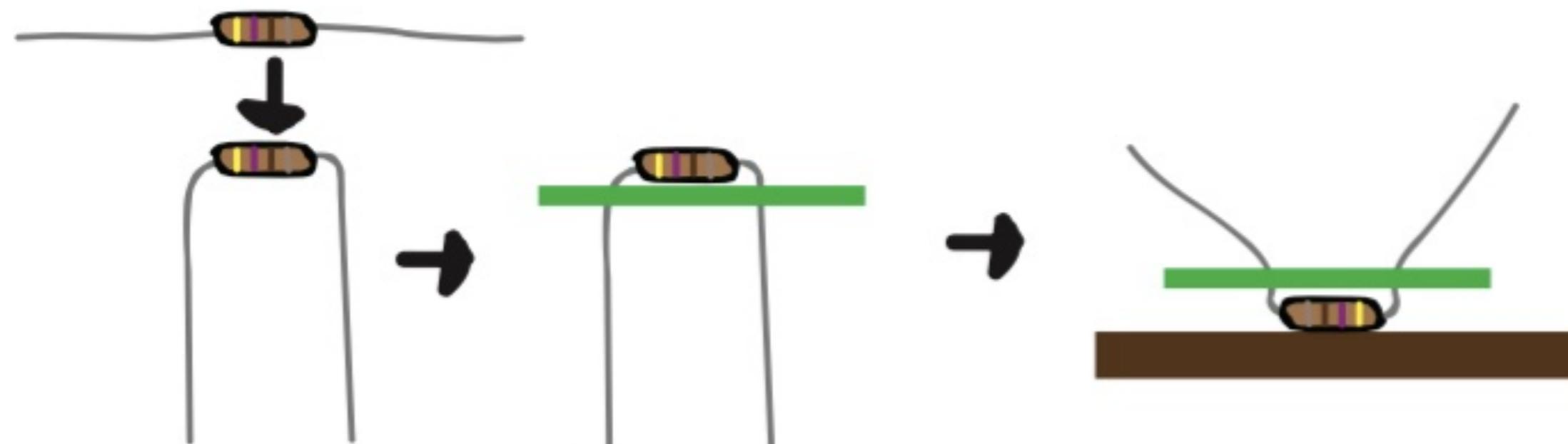
IchigoJam

1,500円~

はじめての「はんだづけ」



1. 部品をみつける（はじめは R1 の抵抗（黄色、むらさき、茶色、金の帯のもの）から！）

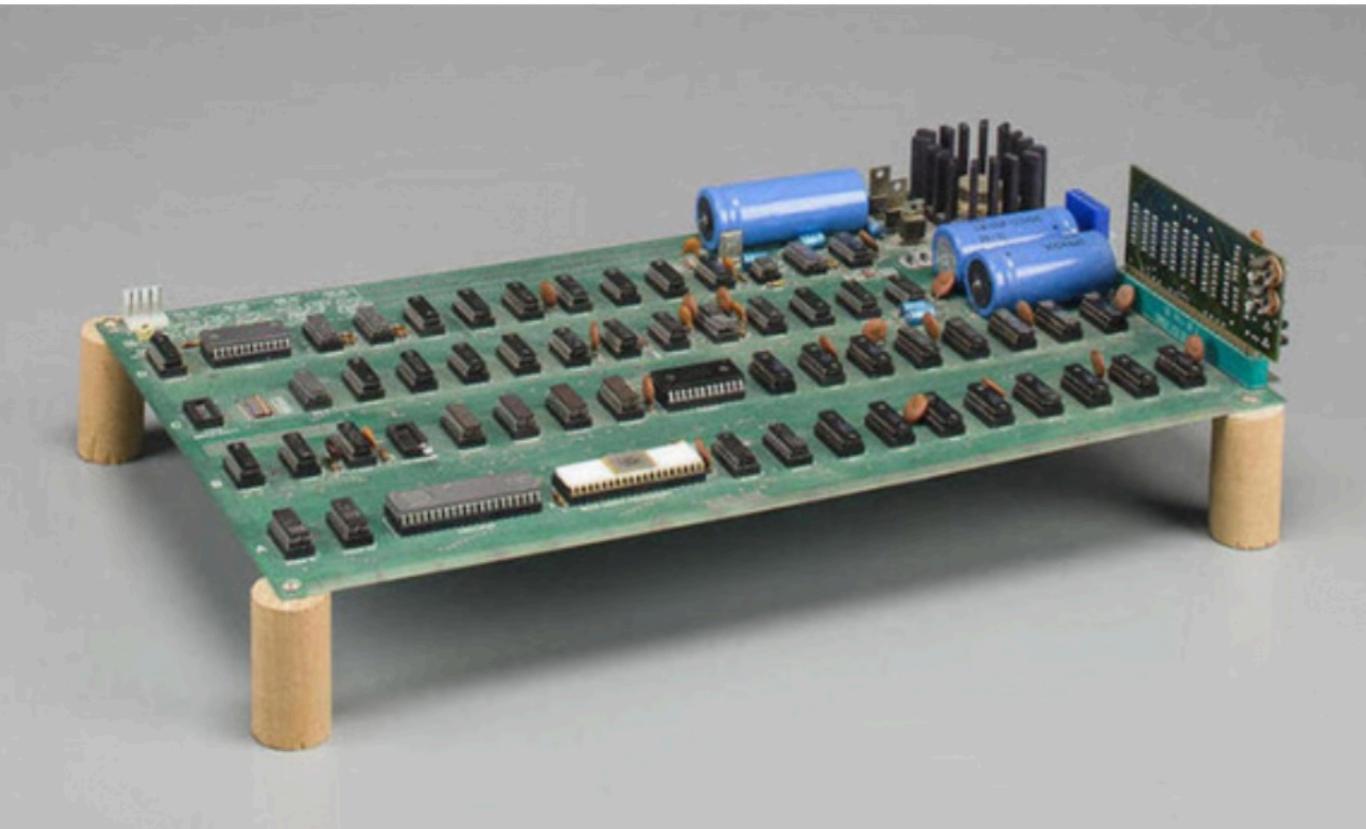


2. ゆびでまげる

3. 基板にさしこむ

★付き部品は向きに注意！

4. 落ちないよう足を少しまげて
うらがえして置く



An Apple I that sold at auction for \$905,000. Source: Bonhams

Apple I (1976年)

iPhoneの会社

Apple社の初製品

じぶんでつくれるパソコン！

Apple I

RAM 4KB

自分で組み立て

BASIC

30万円

楽しい！

IchigoJam

RAM 4KB

自分で組み立て

BASIC

1,500円

楽しい！



Apple I 開発者 - スティーブ・ウォズニアック氏



はんだづけスポット@大阪

共立ものづくり工作室（大阪、日本橋）

BASICで基礎を学んで Pythonなどへステップアップ！

IchigoJam BASIC

Python3

?“Hello!”

print(“Hello!”)

IF A=3:?“YAH!”

```
if a == 3:  
    print("YAH!")
```

組み合わせて使おう

行政データや宇宙データはいくら？

OH

オープンデータ = **自由に使えるデータ**

(営利目的利用、転載、販売、改変など活用自由!)



All rights reserved
(すべての著作権を主張)

クリエイティブ・コモンズ・表示
CC BY
(活用歓迎! 出展だけ表示してね)

多量のデータAIで分析しよう！

最新のAIライブラリ

使用料金はいくら？

OH

多量のデータVRでみよう

最新モバイルVR

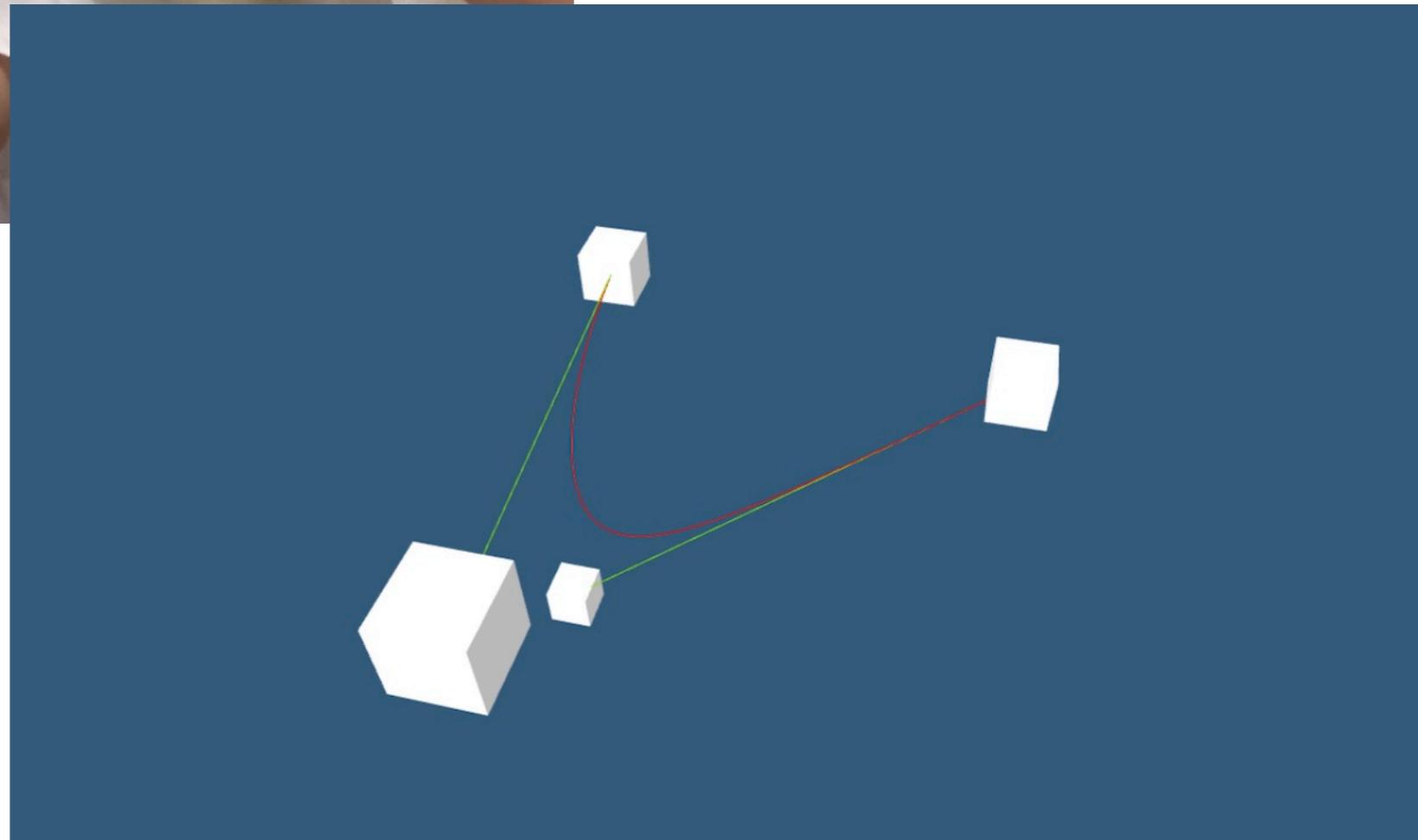
デバイスはいくら？

59,400円



イチオシ
ベジエ曲線 on VR

VRアプリだって
つくれる！



マイクラVR風が たった50行！

WebVR

HTML+JavaScript



```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8"/>
    <title>boxcraft for Oculus Quest - A-Frame</title>
    <script src="https://aframe.io/releases/0.9.1/aframe.min.js">
</script>
</head>
<body>
<script>'use strict'
window.onload = function() {
    const scene = document.querySelector('a-scene')
    const list = []
    handright.addEventListener('triggerdown', function(evt) {
        const p = handright.getAttribute('position')
        const snap = 0.1
        let p2 = {
            x: Math.floor(p.x / snap) * snap + snap / 2,
            y: Math.floor(p.y / snap) * snap + snap / 2,
            z: Math.floor(p.z / snap) * snap + snap / 2,
        }
        let box = document.createElement('a-box')
        box.setAttribute('position', p2)
        box.setAttribute('mixin', 'voxel')
        scene.appendChild(box)
        list.push(box)
    })
    handright.addEventListener('gripdown', function(evt) {
        const b = list.pop()
        if (b)
            scene.removeChild(b)
    })
    handright.addEventListener('bbuttondown', function(evt) {
        const boxes = scene.querySelectorAll('a-box')
        for (let i = 0; i < boxes.length; i++)
            scene.removeChild(boxes[i])
    })
}
</script>
<a-scene>
    <a-assets>
        <a-mixin id="voxel"
            geometry="primitive: box; height: 0.1; width: 0.1; depth: 0.1"
            material="shader: standard">
        </a-mixin>
    </a-assets>
    <a-entity id=handleft oculus-touch-controls="hand: left"></a-entity>
    <a-entity id=handright oculus-touch-controls="hand: right" laser-
    controls raycaster="showLine: true; far: .05" line="color: white; opacity:
    0.75;"></a-entity>
</a-scene>
</body>
</html>
```



鈴木 利器
約1週間前



本日の我家の男子高校生は、鯖江市のオープンデータ活用アプリ「消火栓を探せ」を使って雪に埋もれた消火栓の救出に出かけました♪

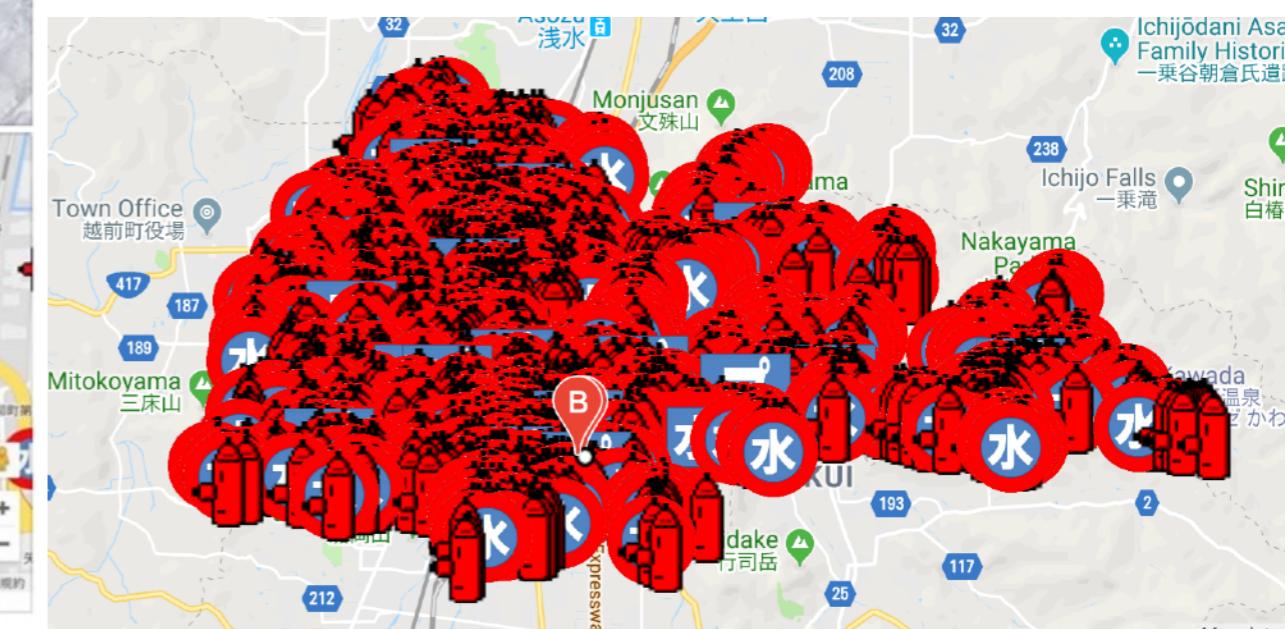
本人いわく、「ゲームアプリ感覚で、宝探しみたいで楽しかった」

近所の3ヶ所の消火栓を無事救出完了(՞՞՞)

楽しく地域の役に立つ活動、休校中の中高生はスマホとスコップ持って外に出よう！... もっと見る



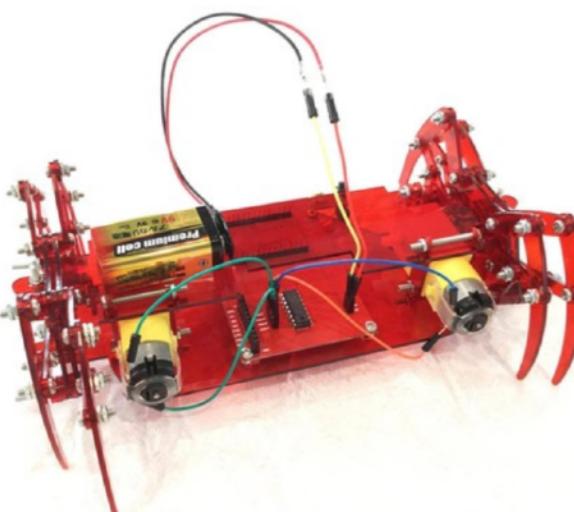
データ × アプリ
まちで遊びながら
社会に貢献！



鯖江で誕生!? 子供起業家

さばえカニロボット

新商品



さばえカニロボット

¥ 8,700

※こちらの価格には消費税が含まれています。
※送料は別途発生いたします。詳細は [こちら](#)
※5,000円以上のご注文で送料が無料になります。

数量 1

カートに入れる

外部サイトに貼る

ツイート シェア 49 通報する



メカ担当：MASAHARU（中2）

基板担当：MISAKI（高2）

Hana道場から子供開発プロダクトの初製品化！

IchigoJam で
I - ト PC を GET ?



CONTEST

◀前のページにもどる



2022/07/20 「PCNこどもプログラミングコンテスト2023」開催決定！

PCNこどもプログラミングコンテスト2023 開催！

今回で第13回目となる一社PCN主催のプログラミングコンテスト「PCNこどもプログラミングコンテスト2023」の開催が決まりました！ プログラミング言語や環境も、テーマも自由！ ゲーム・ツールは勿論、電子工作やロボットと組み合わせたモノとの連携作品などこどもたちのアイデア満載の多種多様な作品を募集します。

詳しい日程や応募方法などは後日公開します。

- 一次審査結果
- 最終審査結果



副賞 ノートPC、3Dプリンター

小中学生 PCNこどもプロコン



<http://pcn.club/contest/>

後援：総務省、文科省、経産省、IT総合室
高専機構、未来の学びコンソーシアム

まなびかたをまなぼう



IchigoJam BASIC リファレンス

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（プログラム変更時もその行で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「\」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 BTN0 END
BTN(数) / ボタン	ボタンが押されれば1、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 "HII"
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]] / ピープ	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 变数,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=	LET A,1

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

コマンド	解説	例
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	比較して以上のように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 变数=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	IN1-9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（IN1,4はブルアップ、IN5-8は切り替え時）	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OUT1-7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（OUT1-4、数2に-1指定でIN5-8へ切り替え）	OUT 1,1
PWM 数1,数2[,数3] / ピーダブリューエム	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

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	CL8DC
On	オクターブ指定 O1C(低音)からO5B(高音)まで 初期値:3	O3C02C
<	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)	C\$DE
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	C'DE

上級コマンド

コマンド	解説	例
CLV / クリア バリアブル	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	入出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カッコ）	?ABS(-2)
[数]	配列 ([0]から[1]まで)を返す	[3]=1
GOSUB 行番号 RETURN	ゴーサブ・リターン	
DEC\$(数)		
#16進数		
HP		

リファレンスをみよう

IchigoJam はじめの一っぽ

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 ぎょうのおわりで、エンターキー

コンピューターに「まで = WAIT (ウェイト)」

WAIT180

WAIT180で3びょうまつ。WAIT60だと？

WAIT60

LEDを1びょうひからせる (: コロンでつなぐ)

LED1 : WAIT60 : LED0

カーソルキーのうえキーを2かいおす
みぎキーを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
18とうち、さいごにエンターキー

LED1 : WAIT180 : 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)

1 LED1 : WAIT180
2 LED0 : WAIT180
3 GOT01
RUN

ひだりうえのESC (エスケープ) キーでストップ
RUNのかわりに、F5キーでもOK！

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といい？
かえたら、かえたぎょうで、エンターキー

つくったプログラムを、ほぞんしよう (0~3の4つ)

SAVE0 セーブ、F3キー、0でもOK！

スイッチをきっても、もとどおり

LOAD0 ロード、F2キー、0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

 キーのうえにあるもじはシフトキーをおしながらおす



かわくだりゲーム（カーソル左右でよけろ！）

```

10 CLS : X=15
20 LC X,5 :"0"
30 LC RND(32),23 :"*"
35 WAIT 3
36 X=X-BTN(28)+BTN(29)
40 IF SCR(X,5)=0 GOTO 20

```

※ 0を◎にかえる → Altキーをおしながら C

はんのうそくどゲーム

(LEDひかったらすぐ、スペースキー)

```

10 LED0
20 WAIT RND(180)+60
30 LED1:CLT
40 IF BTN(32)=0 GOTO 40
50 ?TICK()

```

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（タイミングよくキーをおす）

```

10 Y=0
20 CLS
30 LC 4,15 :"%"
40 LC 5,Y :"0"
50 IF INKEY() 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（キーをいろいろおすと？）

```

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

```



BASICでプログラミング！

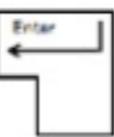
こどもパソコン IchigoJam



<http://ichigojam.net/>

ひかりをあやつるLED（エルイーディー）

IchigoJamのスイッチを入れて、LED1（エル、イー、ディー、い
ち）とキーボードからうちこんで、Enter（エンター）キーをおして
みよう。（Enterキーは、みぎのようなおおきなキー）



LED1←

「OK（オーケー）」とでて、IchigoJamのLEDがひかったら、だいせいこう！

LED0（エル、イー、ディー、ゼロ）、エンターでけせる。

LED0←

キーボードのまんなかしたにあるなにもかいてない大きなキ
ーはスペースキー。LED1（エル、イー、ディー、スペ
ース、いち）、エンターと、スペースはあってもなくてもOK。

LED 1←

RED0（アール、イー、ディー、ゼロ）、エンターで、LEDはきえるかな？

RED0←

「Syntax error（シンタックス・エラー）」とでて、きえません。
などまちがえても、おこらないのがコンピューター。

ABD（エー、ビー、ディー）とうってみましょう。エンターキーはおさない。

ABD←

ここで Backspace（バックスペース）キー をひとおし。
Backspaceキーは、キーボードの右上のほうにあります。



AB←

ひともじけせました。これで、うちまちがいもこわくない。

やってみよう！

1. ABCDEFGとかいてみよう
2. LEDをけしてみよう
3. すばやくLEDをつけてけそう
4. BEEPとかいてエンターおしてみよう
5. CLSとかいてエンターおしてみよう

IchigoJamプリント

A5印刷対応ネット教材

<http://ichigojam.net/print/>

うごかしてみよう

IchigoJam web by WebAssembly X +

https://fukuno.jig.jp/app/IchigoJam/... 🔍 ⭐ ⓘ

IchigoJam web

by WebAssembly

```
IchigoJam BASIC 1.3.0 web jig.jp
OK
10 CLS:X=15
20 LC X,5:?"
30 LC RND(32),23:?**"
40 GOTO 20
```

esc / ctrl-c export import full screen audio on

```
10 CLS:X=15
20 LC X,5:?"
30 LC RND(32),23:?**"
40 GOTO 20
```

[チュートリアル / IchigoJam BASIC リファレンス \(English\)](#)

【IchigoJamプログラミング、ネット教材】

[チュートリアル付き IchigoJam web](#)
[はじめのいっぽ&ミニゲームズ](#)
[IchigoJamプリント](#)
[IchigoJamプログラミング入門](#)
[ダンブンゲームズ by PCN](#)

IchigoJam web : [IchigoJam \(blog fukuno.jig.jp\)](#)
API: [WebAssembly / WASM](#)

IchigoJam web

おうちのパソコンでもうごく
IchigoJam

<https://fukuno.jig.jp/app/IchigoJam/>

ウデマエ

ひろうしよう

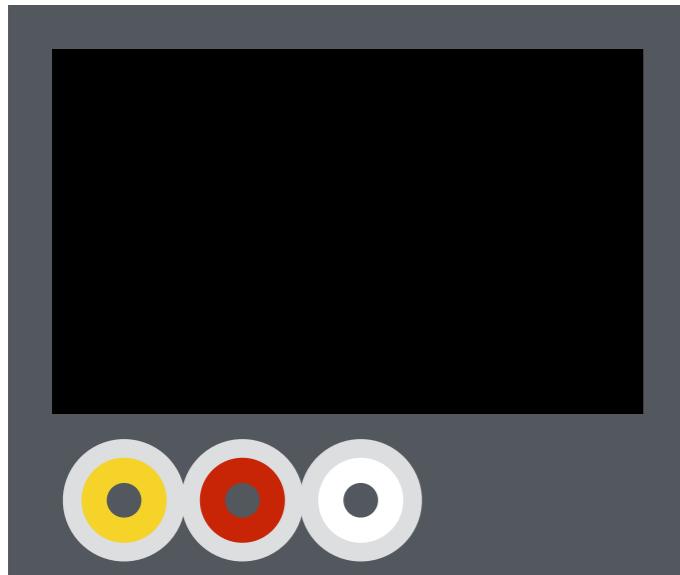
大人も楽しく一緒に学ぶ「地域ICTクラブ」



83歳プログラマー
若宮さん@鯖江 Hana道場

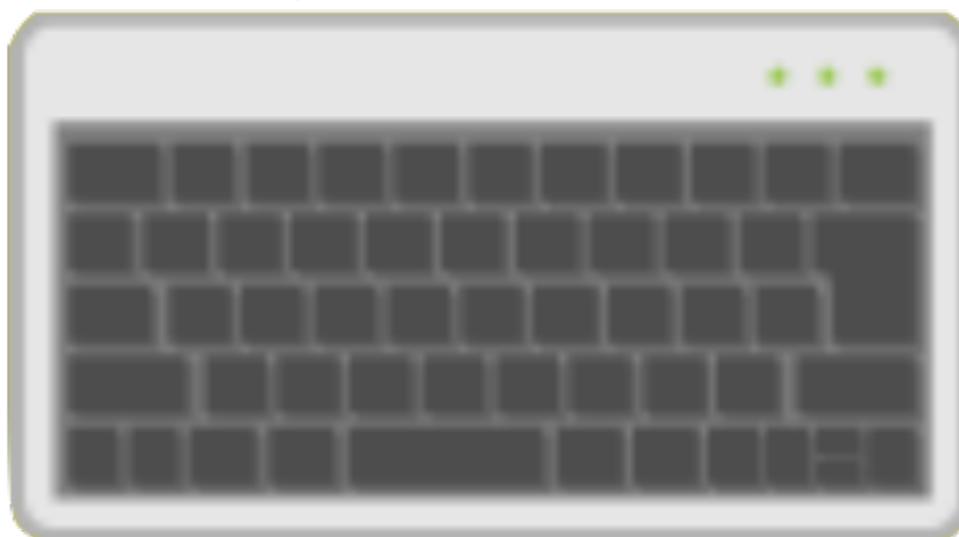
IchigoJam にひつような もの

ご家庭のテレビ

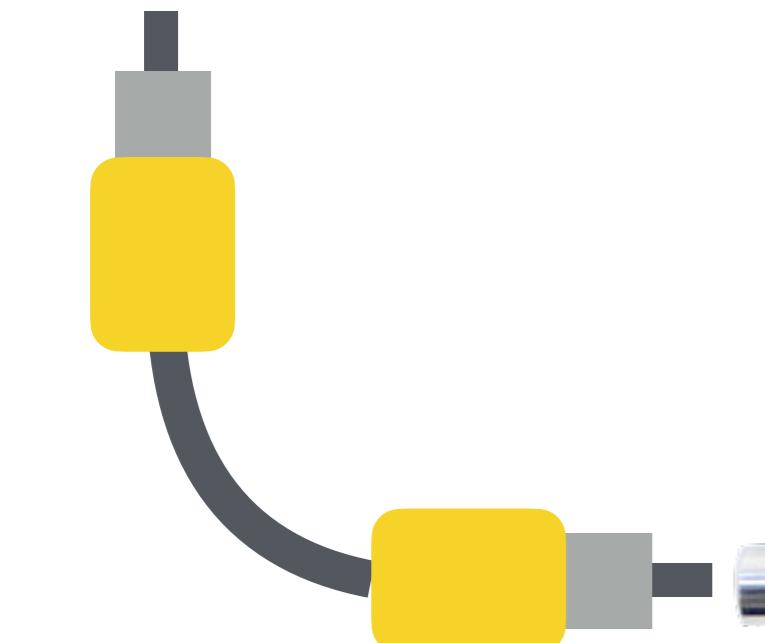


1. サンワサプライ 有線USBキーボード
SKB-L1UBK Amazonにて676円

<https://www.amazon.co.jp/dp/B005LL9J9G/>



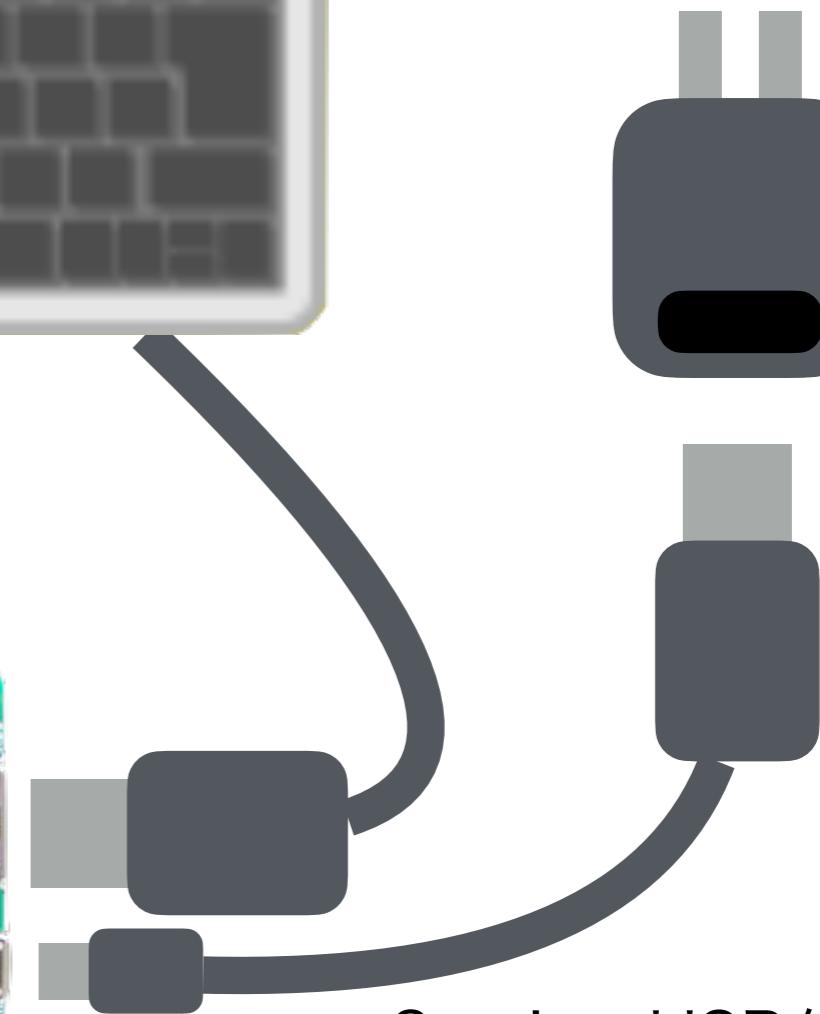
2. USB ACアダプター
DAISOにて324円



4. コンポジットケーブル
黄赤白、いずれか1本でOK
DAISOにて108円



3. microUSBケーブル
DAISO/Seriaにて108円



困った時は、ネットかHana道場で聞こう！

Facebookグループ 「IchigoJam-FAN」

メンバー 2500人+ (or Twitter #IchigoJam)

Facebook IchigoJam-FAN

IchigoJam-FAN 公開グループ

情報 ディスカッション

メンバー イベント 動画 写真 ファイル グループインサイト グループのモデレーション

このグループを検索

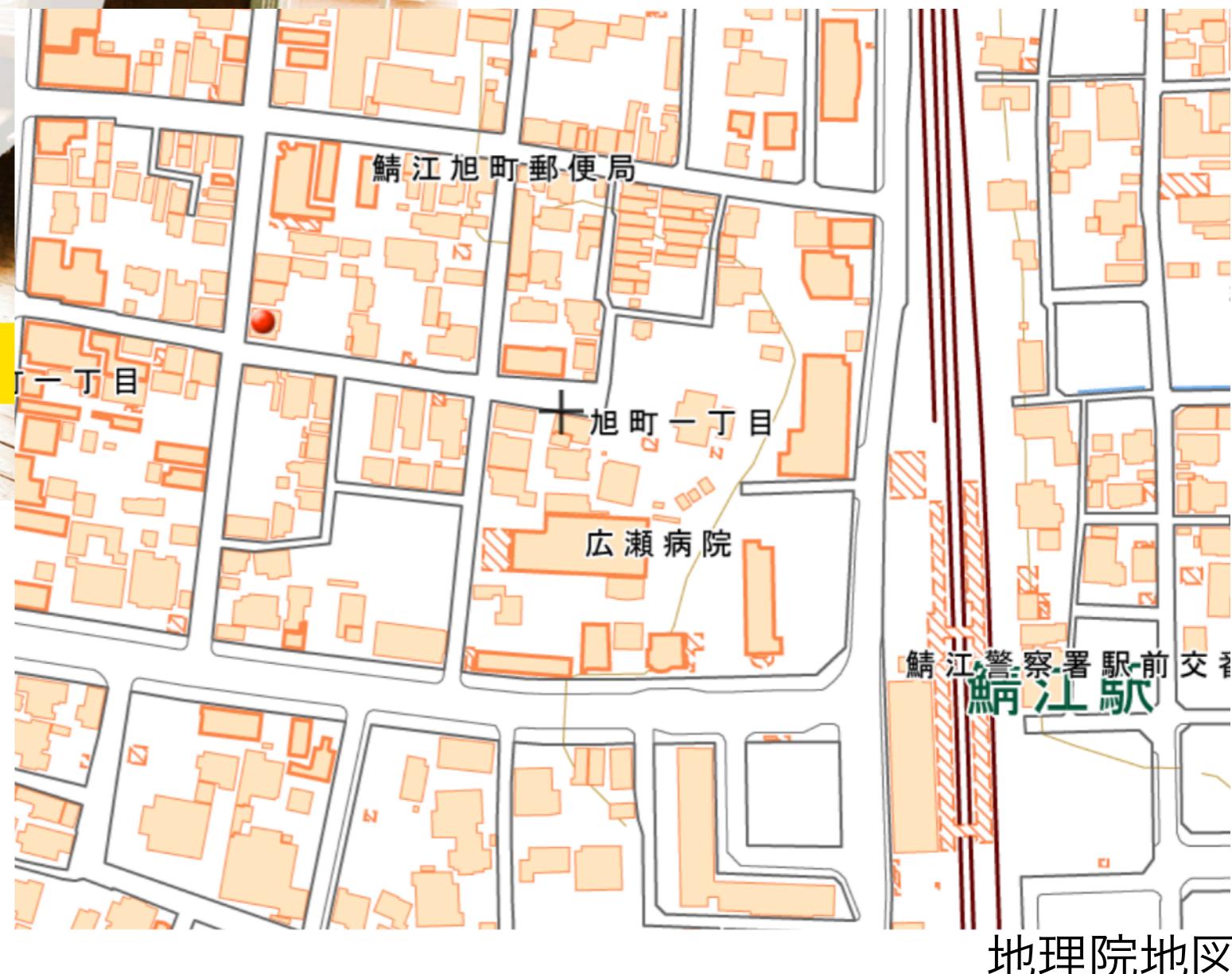
QR code



はんだづけもできるよ
IchigoJam
スペシャリストたくさん

こどもも大人も
ITを楽しく学ぶ
Hana道場

<https://hanadojo.com/>



地理院地図



IchigoJam検定とは

IchigoJam検定は、IchigoJamでのプログラミングの理解度・習熟度を測る検定試験です。

近年、総務省から発表されている「若年層におけるプログラミング教育の普及推進事業」にも伴い、プログラミング教育への関心が高まっている背景のもと、プログラミング教育の認知度向上およびこどもたちのステップアップ意欲の育成を目的とし、知識やスキルを計るだけでなく、IchigoJamを利用したプログラミング教室などの教育カリキュラム基準にご活用いただけます。

Hana道場ではIchigoJam検定も受験できます！



だい 6 かい
第6回 越前がに
えちぜん
ロボコン

2022 12.17 SAT

ショッピングシティベル あじさいホール

ふくいけん ふくいしはなんどうみなみ ちょうめ
福井県 福井市 花堂南2丁目16-1



コンテスト概要



ルールについて



コースの作り方





@taisukef

<https://fukuno.jig.jp/>



株式会社 jig.jp 会長 福野泰介 @taisukef / Facebook
(IchigoJam発明者、デジタル庁オープンデータ伝道師)

