

# 次世代 キッズプログラミング教室

第1回目：2020年1月26日

micro:BITボードについての基礎、基本表示、センサーの利用

第2回目：2020年2月2日

外部出力、無線通信、おまけ

第3回目：2020年2月24日

総合演習（復習）、みんなが作ったものの発表など

(はじめに)

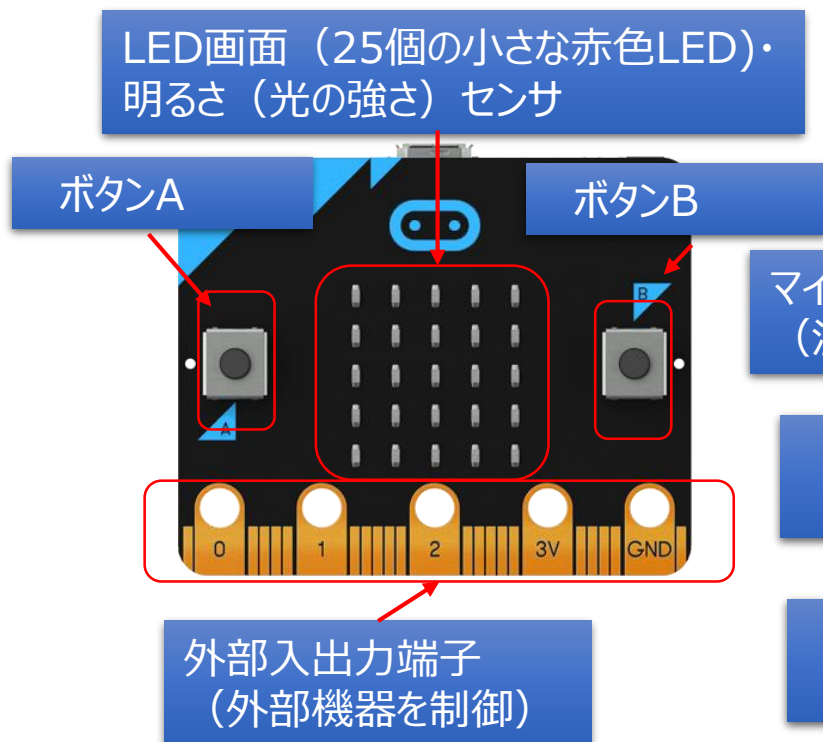
# micro:bitを観察しよう！

【micro:bitとは】(公式HP : <https://microbit.org/ja/guide/>)

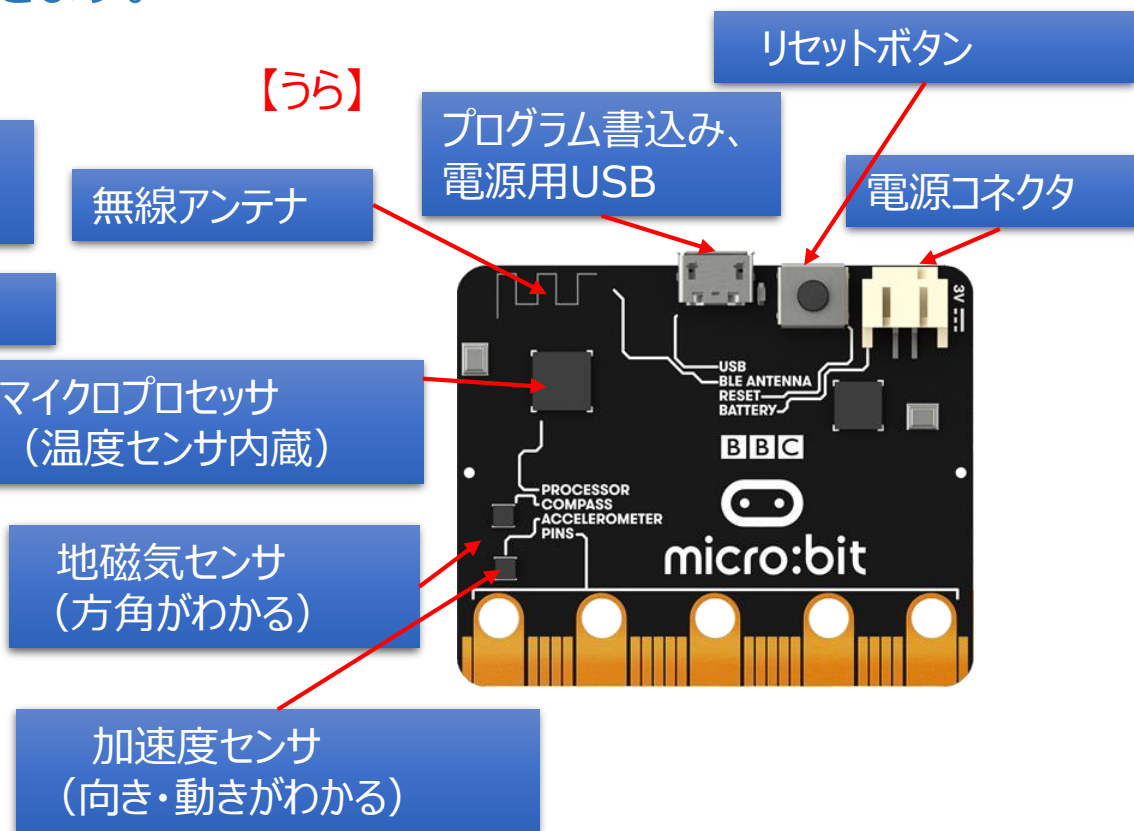
micro:bitは、2016年にイギリスの公共放送局であるBBCが中心となって開発した小型の教育用コンピューターです。

25個の小さな赤色LED・2個のスイッチ・さまざまなセンサー・無線通信機能などを搭載し、プログラムで制御することができます。

【おもて】



【うら】

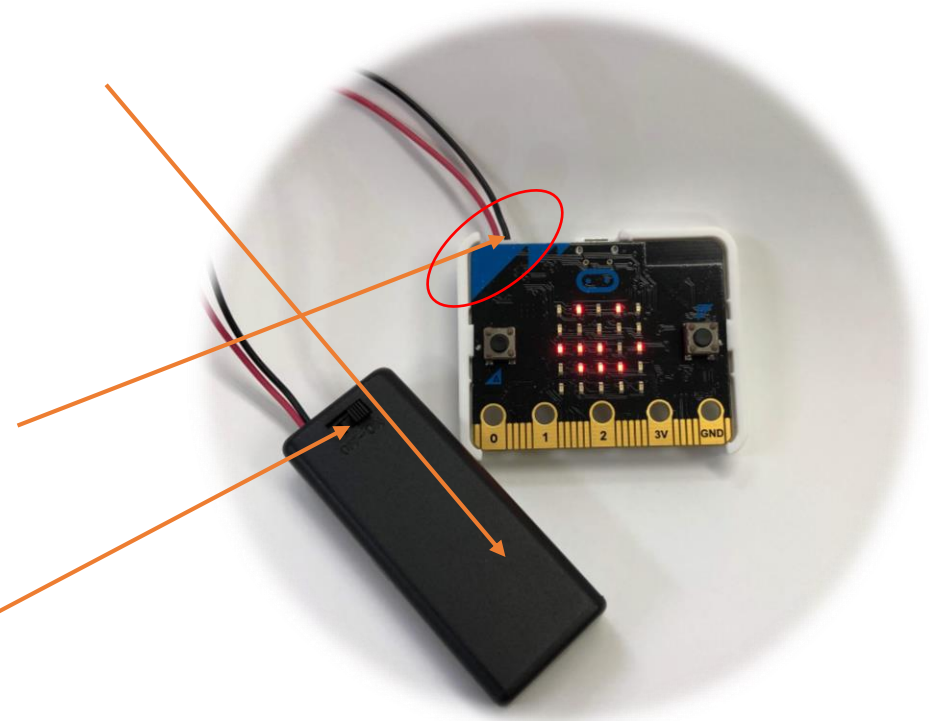


# micro:bitキットをセットアップしよう！

① 電池ボックスに電池をいれよう

② 電源コネクタにつなごう

③ 電池ボックスをON



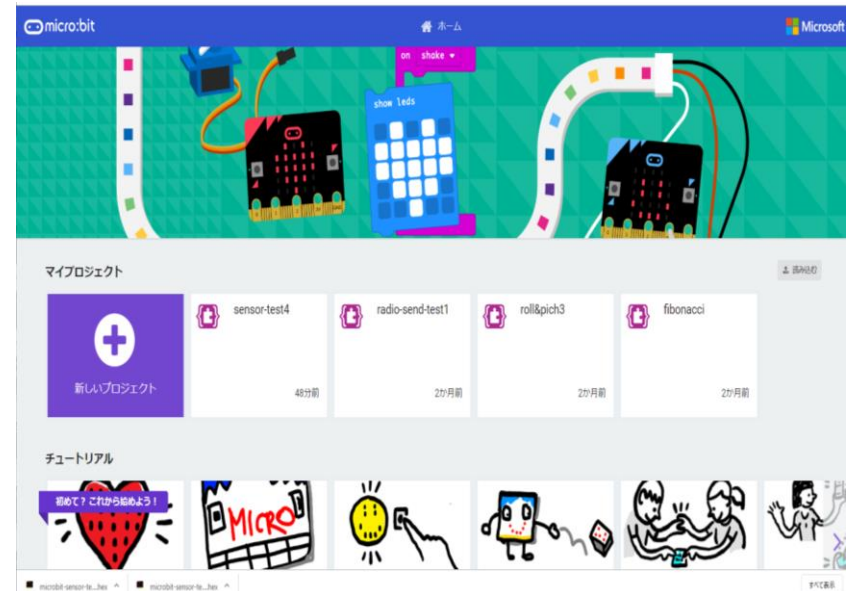
micro:bitは動きだしたかな？

# それでは、プログラミングを始めよう！

## 【準備：MakeCodeエディタを立ち上げます】

### 【パソコン（WindowsやMac）の場合】

- ①インターネットに接続していることを確認します。
- ②ブラウザ（Chrome やSafariなど）を立ち上げます。
- ③<http://makecode.microbit.org/>にアクセスします。



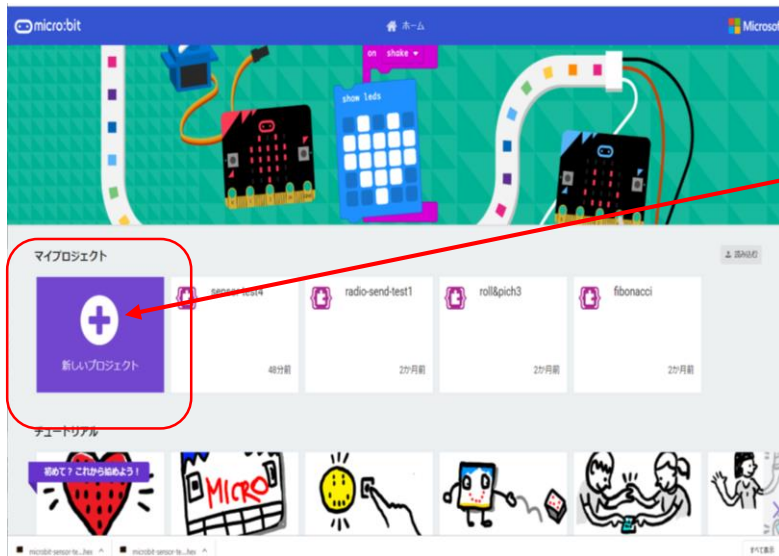
### 【Windows 10 バージョン 10240.0 以降のパソコンをお持ちなら】

- ①Microsoftから無料プログラムソフト「**MakeCode for micro:bit**」をダウンロードしインストールしておくと便利です。

# 【ブロック・プログラミング】

①ここをクリックし「新しいプロジェクト」を作ります

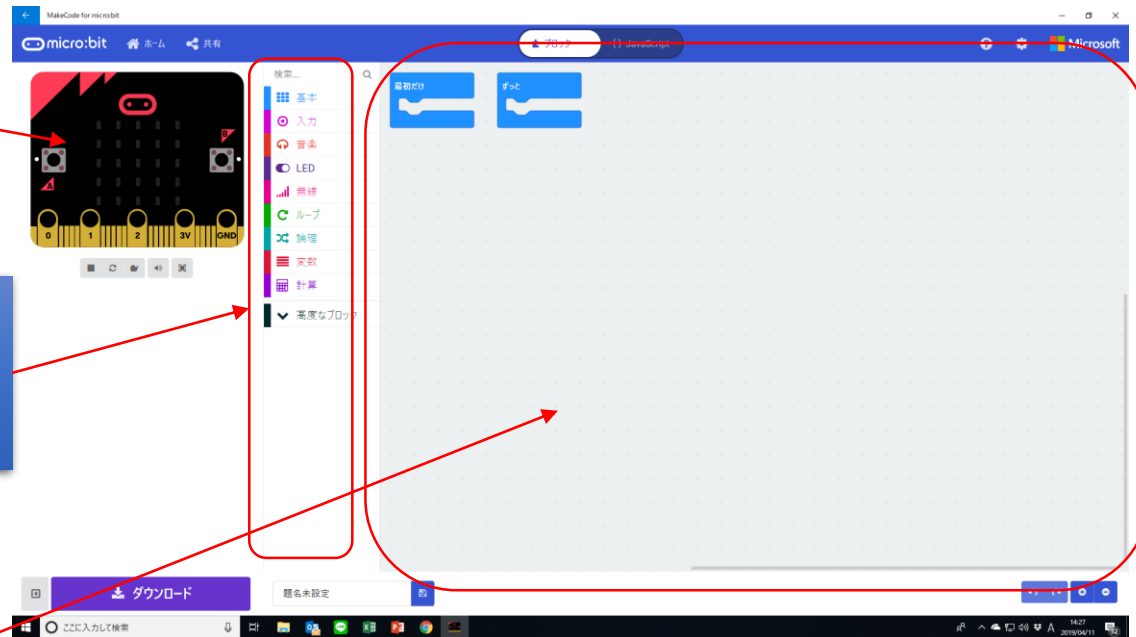
②ツールボックスからプログラムブロックを選んで、ワークスペースにドラッグし、ならべて、動作させたいプログラムを作っていきます



micro:bitエミュレータ  
(micro:bitの動作が確認できる)

ツールボックス  
(いろいろな命令ブロックメニューが  
そろっている)

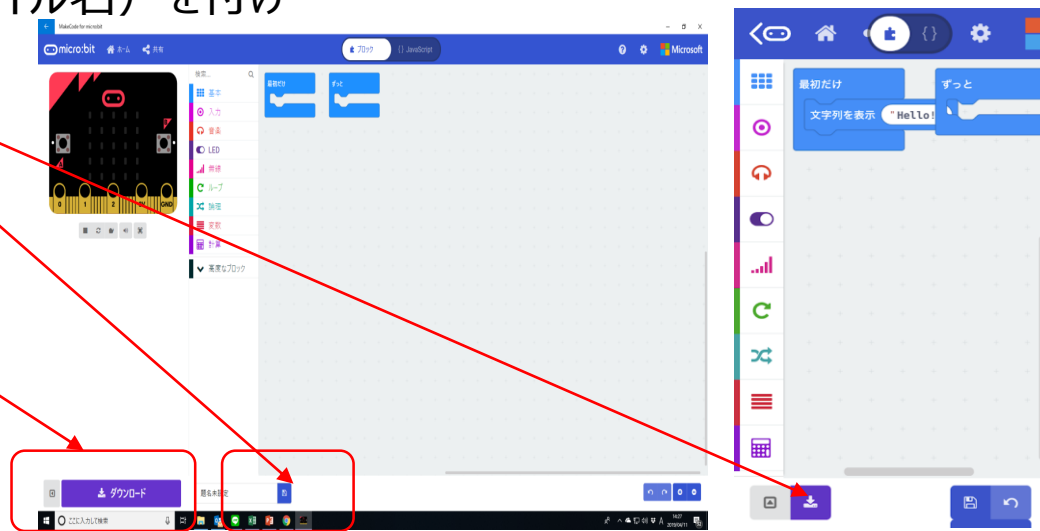
ワークスペース  
(命令ブロックをならべてプログラム  
を組むところ)



# 【プログラムを保存します】

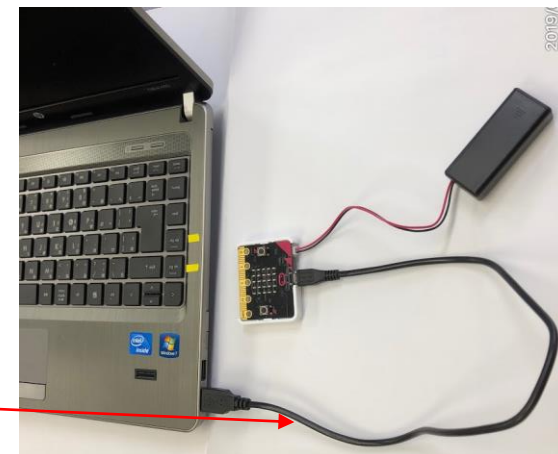
③できたプログラムはプロジェクト名（ファイル名）を付けて、ファイルに保存します

○Windows10のMakeCodeアプリの場合  
はダウンロードをクリック



# 【micro:bitとパソコンを接続しプログラムをmicro:bitにダウンロード（書込む）準備をします】

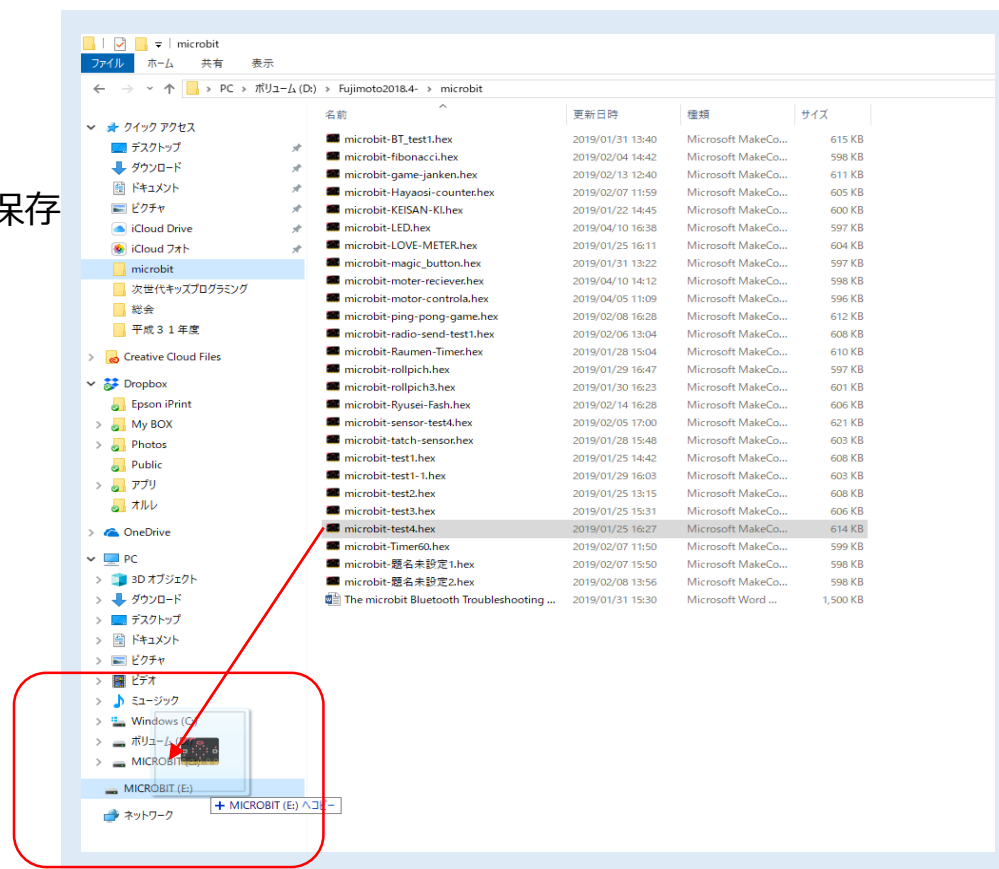
④プログラムが書けたら、USBケーブルでパソコンとmicro:BITを接続する



# 【micro:bitに書き込み、うまく動くかチェックします】

## ⑤micro:BITにプログラムを書き込む

○Windowsはエクスプローラ、MacはFinderを使い、保存したプログラムファイルをmicro:BITにドラッグ・ドロップ



【micro:bitの動作をたしかめて、思い通りに動かなければ②にもどってプログラムをチェック】

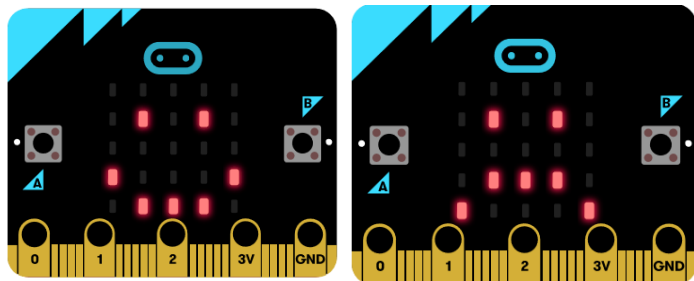
# プログラミングを始めてみよう

## 目標

まずは、プログラムを作りmicro:bitを動かしてみよう

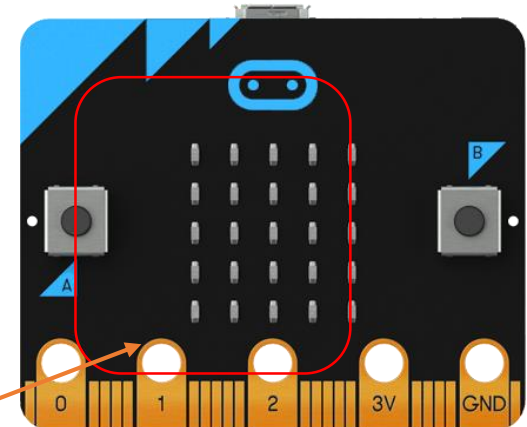
## やりたいこと

用意されている顔文字（アイコン）を交互にLED画面に表示させる



「やりたいこと」を分解してみる

LED画面



②アイコンを表示させる

一番目のアイコンを表示

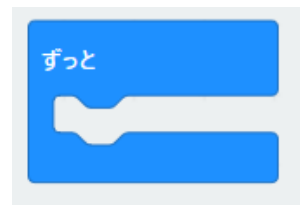
二番目のアイコンを表示

①繰り返して



## プログラミング・ブロックで考えると

分解した図をプログラミング・ブロックで考えてみると、「ずっと」と「アイコンを表示」ブロックを組合わせて使うことになるよ



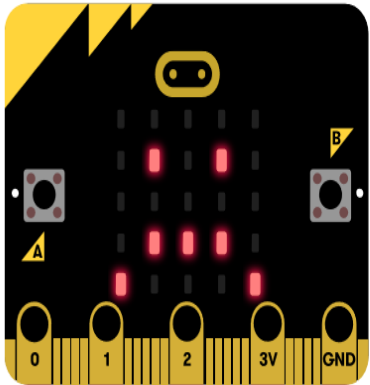
同じことをずっと繰り返すブロックを使って、処理を繰り返させる



「アイコンを表示」ブロックを使って顔文字を表示させる

それでは、早速エディタでプログラミングしてみよう！！

- ①「[makecode.microbit.org](https://makecode.microbit.org)」へアクセスする。またはアプリを立ち上げる
- ②「新しいプロジェクト」をクリック
- ③エディタ上でブロックプログラミング！ 出来たら名前を付けて保存
- ④できたプログラムファイルをmicro:bitにダウンロード（書込む）



検索...

基本

入力

音楽

LED

無線

ループ

論理

変数

計算

高度なブロック

最初だけ

ずっと

アイコンを表示

アイコンを表示

画面ウィンドウの表示例

ここで動きをチェック

ダウンロード

題名未設定



# わくわくプログラミング①

**目標** 自分のネームプレート（名札）を作ってみよう

**やりたいこと**

micro:bitのLED画面に、電光掲示板のように繰り返して自分の名前をカタカナで表示させる

「やりたいこと」を分解してみる（タロウの場合）

②自分の名前をカタカナで表示

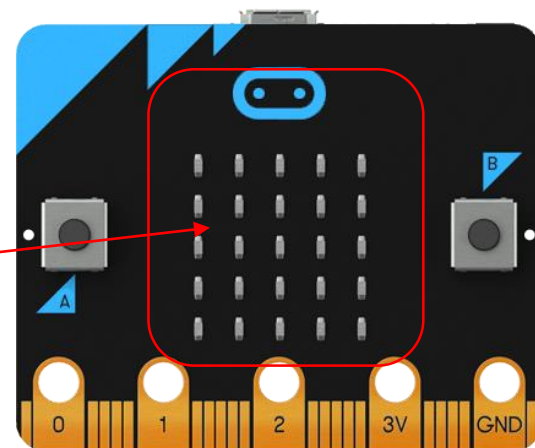
一文字目（タ）を表示させる

二文字目（ロ）を表示させる

三文字目（ウ）を表示させる

①繰り返して

LED画面



## プログラミング・ブロックで考えると

分解した図をプログラミング・ブロックで考えてみると、「ずっと」と「LED画面に表示」を使うことになるよ



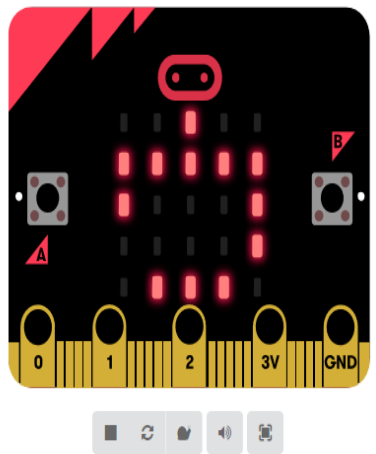
→ 同じことを**ずっと**繰り返すブロックを使って、繰り返させる



→ **LED画面に表示**させるブロックを使って、白いところを点灯させる

それでは、エディタ画面でプログラミングしてみよう！！

- ①「[makecode.microbit.org](https://makecode.microbit.org)」へアクセス、またはアプリを立ち上げる
- ②「新しいプロジェクト」をクリック
- ③エディタ上でブロックプログラミング！ 出来たら名前を付けて保存
- ④できたプログラムファイルをmicro:bitにダウンロード（書込む）



- 検索...
- 基本
  - 入力
  - 音楽
  - LED
  - 無線
  - ループ
  - 論理
  - 変数
  - 計算
  - 高度なブロック



画面ウィンドウの表示例

## チャレンジ①！！

「基本」メニューの中の「アイコンを表示」ブロックを使って、名前の前後に好きなアイコンを表示させてみよう

## チャレンジ②！！

「基本」メニューの中の「文字列を表示」ブロックを使って、名前をローマ字で表示させて、表示の出方の違いを見てみよう  
？文字列に日本語を入れたらどうなる？

## チャレンジ③！！

Aボタンを押したときだけ、名前を表示するように変更してみよう

→ 「ずっと」ブロックの代わりに、「入力」メニューにある「ボタンAが押されたとき」ブロックを使うよ



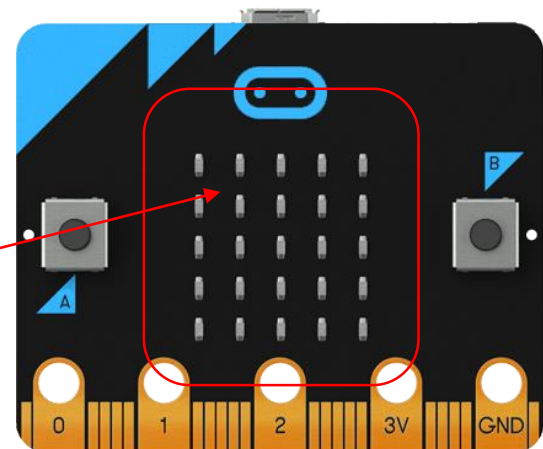
# わくわくプログラミング②

**目標** 部屋などの明るさを測ってみよう

**やりたいこと**

micro:bitのLED画面センサがとらえる明るさを、LED画面に  
表示させる

LED画面  
明るさセンサ



「やりたいこと」を分解してみる

②明るさをLEDに表示させる

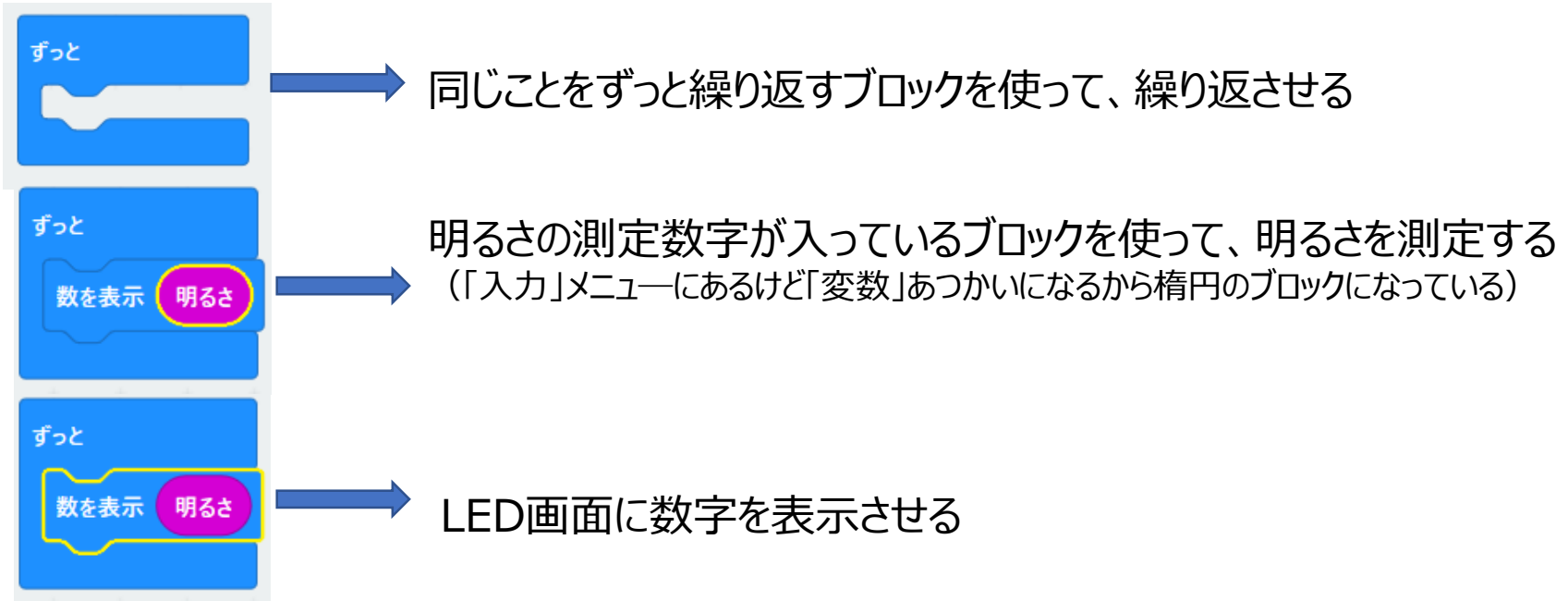
明るさの測定（入力）

LED画面に明るさを表示

①繰り返して

## プログラミング・ブロックで考えると

分解した図をプログラミング・ブロックで考えてみると、「ずっと」と「明るさ」「数を表示」を使うことになるよ



それでは、エディタ画面でプログラミングしてみよう！！

- ①「[makecode.microbit.org](https://makecode.microbit.org)」へアクセス、またはアプリを立ち上げる
- ②「新しいプロジェクト」をクリック
- ③エディタ上でブロックプログラミング！ 出来たら名前を付けて保存
- ④できたプログラムファイルをmicro:bitにダウンロード（書き込む）





計算

数を表示 明るさ



## チャレンジ①！！

いろいろな場所で明るさを測ってみよう。どんな数字になるかを観察しよう。

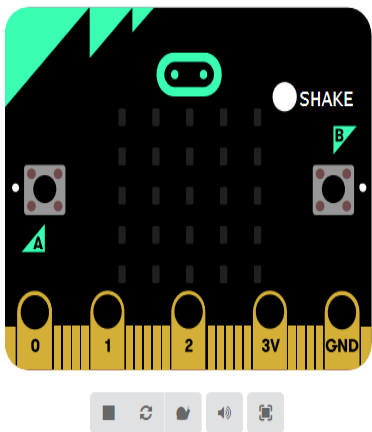
## チャレンジ②！！

micro:bitには「温度センサ」や「方角センサ（コンパス）」「加速度センサー」なども入っています。明るさの代わりに温度や方角を測定して、どんな数字になるかを観察しよう。

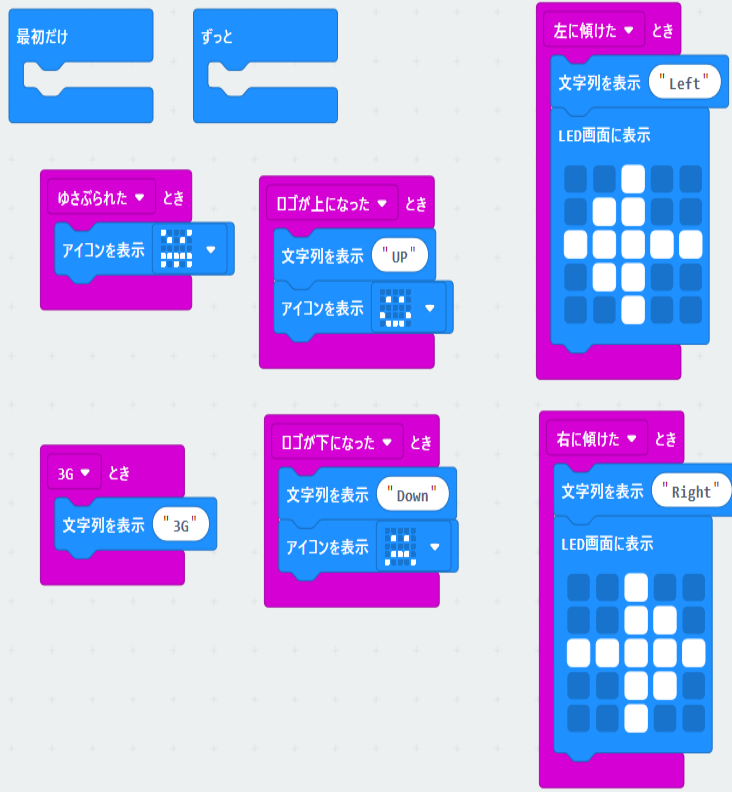
## チャレンジ③！！

「加速度センサー」を使って何か表示させてみよう





- 検索...
- 基本
  - 入力
  - 音楽
  - LED
  - 無線
  - ループ
  - 論理
  - 変数
  - 計算
  - 高度なブロック



加速度センサーを使ったプログラム例

# わくわくプログラミング③

## 外部機器の制御

### 目標

明るさ検知懐中電灯を作ろう

### やりたいこと

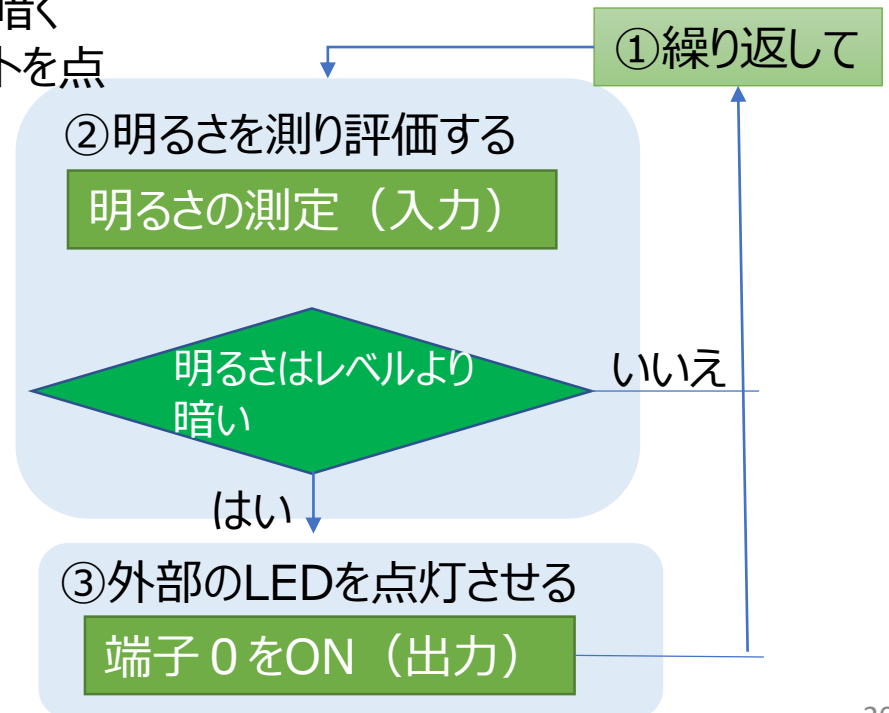
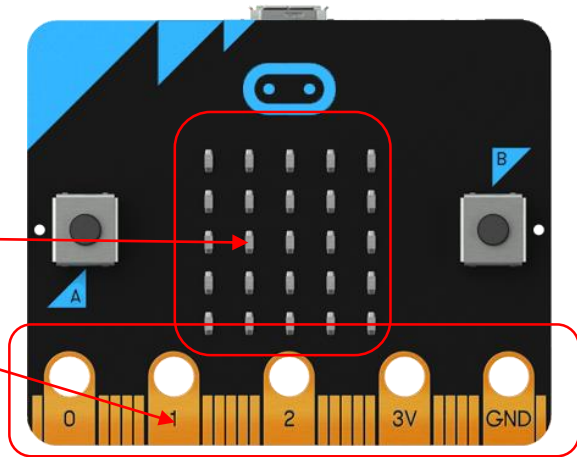
micro:bitの明るさセンサを利用し、あるレベルに暗くなったら外部出力端子を使って接続したLEDライトを点灯させる

「やりたいこと」を分解してみる

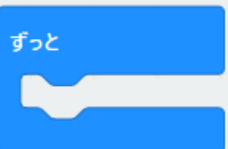
論理分岐や外部  
入出力など少し高  
度になるよ

明るさセンサ

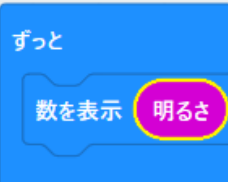
外部出力端子



## プログラミング・ブロックで考えると



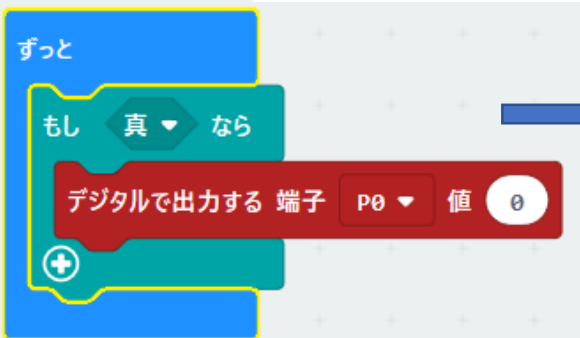
→ 同じことを**ずっと**繰り返すブロックを使って、繰り返させる



→ **明るさ**の測定数字が入っているブロックを使って、明るさを測定する  
(「入力」だけど「変数」になるから楕円のブロックになっている)



→ 「**論理式**」メニューの「もし～なら」ブロックで判定、分岐させる



→ 「**高度なブロック**」にある「**入出力**」メニューの「**デジタルで出力する**」で、外部LEDを点灯させる

それでは、エディタ画面でプログラミングしてみよう！！

MakeCode for micro:bit

micro:bit ホーム 共有

ブロック JavaScript

検索...

基本  
入力  
音楽  
LED  
無線  
ループ  
論理  
変数  
計算  
高度なブロック  
関数  
配列  
文字列  
ゲーム  
画像  
入出力端子  
シリアル通信  
制御  
拡張機能

128

0 1 2 3V GND

ずっと  
もし 明るさ < 128 なら  
デジタルで出力する 端子 P0 値 1  
でなければ  
デジタルで出力する 端子 P0 値 0

画面ウィンドウの表示例

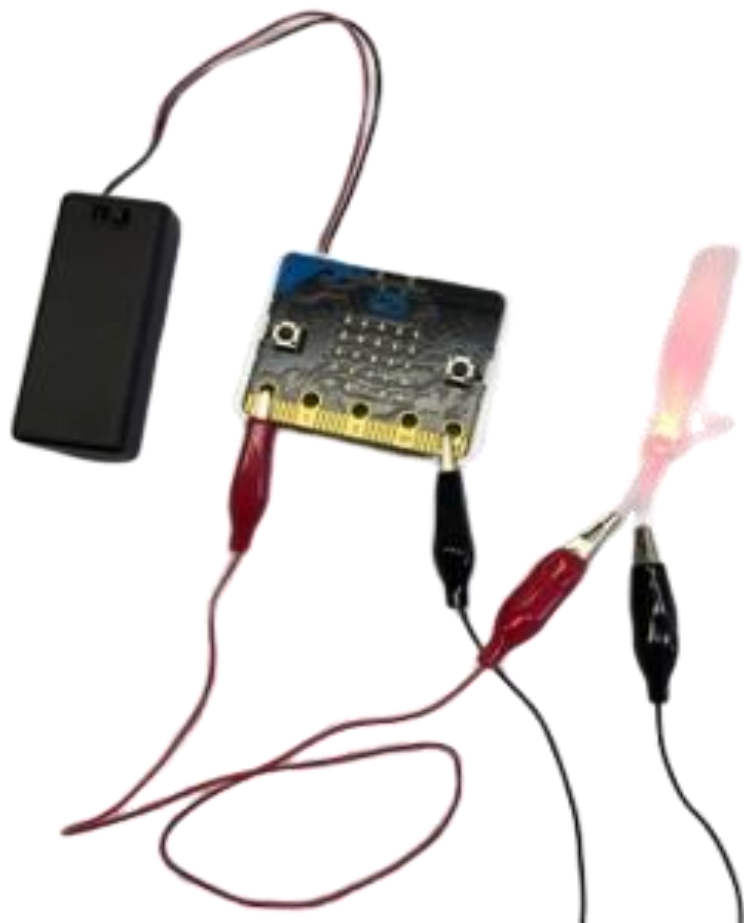
ダウンロード

題名未設定

ここに入力して検索

2019/04/12 15:11

それでは、micro:bitとLEDをミノムシクリップでつないでみよう



① ミノムシクリップ（ワニぐち）で、0 番端子とLEDの長い足、GND端子とLEDの短い足をつなぐ

② micro:bitの電源スイッチをON

手でmicro:bitをおおうとLED点灯

LEDの足を逆につなぐと点灯するかな？  
どうなるか試してみよう

## チャレンジ①！！

光らせる強さを変えて調光ランプにしてみよう（やり方は自由に）

## チャレンジ②！！

スピーカーをつないでメロディーを鳴らしてみよう。

## チャレンジ③！！

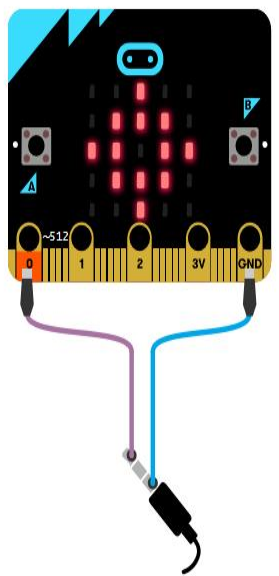


## チャレンジ①のプログラム例



## チャレンジ②のプログラム例





- 検索...
- 基本
  - 入力
  - 音楽
  - LED
  - 無線
  - ループ
  - 論理
  - 変数
  - 計算
  - 高度なブロック

```
for (let i = 0; i < 10; i++) {
  playTone(440, 1);
  playTone(480, 1);
  playTone(520, 1);
  playTone(560, 1);
  playTone(600, 1);
  playTone(640, 1);
  playTone(680, 1);
  playTone(720, 1);
  playTone(760, 1);
  playTone(800, 1);
  playTone(840, 1);
  playTone(880, 1);
  playTone(920, 1);
  playTone(960, 1);
  playTone(1000, 1);
  playTone(1040, 1);
  playTone(1080, 1);
  playTone(1120, 1);
  playTone(1160, 1);
  playTone(1200, 1);
  playTone(1240, 1);
  playTone(1280, 1);
  playTone(1320, 1);
  playTone(1360, 1);
  playTone(1400, 1);
  playTone(1440, 1);
  playTone(1480, 1);
  playTone(1520, 1);
  playTone(1560, 1);
  playTone(1600, 1);
  playTone(1640, 1);
  playTone(1680, 1);
  playTone(1720, 1);
  playTone(1760, 1);
  playTone(1800, 1);
  playTone(1840, 1);
  playTone(1880, 1);
  playTone(1920, 1);
  playTone(1960, 1);
  playTone(2000, 1);
  playTone(2040, 1);
  playTone(2080, 1);
  playTone(2120, 1);
  playTone(2160, 1);
  playTone(2200, 1);
  playTone(2240, 1);
  playTone(2280, 1);
  playTone(2320, 1);
  playTone(2360, 1);
  playTone(2400, 1);
  playTone(2440, 1);
  playTone(2480, 1);
  playTone(2520, 1);
  playTone(2560, 1);
  playTone(2600, 1);
  playTone(2640, 1);
  playTone(2680, 1);
  playTone(2720, 1);
  playTone(2760, 1);
  playTone(2800, 1);
  playTone(2840, 1);
  playTone(2880, 1);
  playTone(2920, 1);
  playTone(2960, 1);
  playTone(3000, 1);
  playTone(3040, 1);
  playTone(3080, 1);
  playTone(3120, 1);
  playTone(3160, 1);
  playTone(3200, 1);
  playTone(3240, 1);
  playTone(3280, 1);
  playTone(3320, 1);
  playTone(3360, 1);
  playTone(3400, 1);
  playTone(3440, 1);
  playTone(3480, 1);
  playTone(3520, 1);
  playTone(3560, 1);
  playTone(3600, 1);
  playTone(3640, 1);
  playTone(3680, 1);
  playTone(3720, 1);
  playTone(3760, 1);
  playTone(3800, 1);
  playTone(3840, 1);
  playTone(3880, 1);
  playTone(3920, 1);
  playTone(3960, 1);
  playTone(4000, 1);
  playTone(4040, 1);
  playTone(4080, 1);
  playTone(4120, 1);
  playTone(4160, 1);
  playTone(4200, 1);
  playTone(4240, 1);
  playTone(4280, 1);
  playTone(4320, 1);
  playTone(4360, 1);
  playTone(4400, 1);
  playTone(4440, 1);
  playTone(4480, 1);
  playTone(4520, 1);
  playTone(4560, 1);
  playTone(4600, 1);
  playTone(4640, 1);
  playTone(4680, 1);
  playTone(4720, 1);
  playTone(4760, 1);
  playTone(4800, 1);
  playTone(4840, 1);
  playTone(4880, 1);
  playTone(4920, 1);
  playTone(4960, 1);
  playTone(5000, 1);
  playTone(5040, 1);
  playTone(5080, 1);
  playTone(5120, 1);
  playTone(5160, 1);
  playTone(5200, 1);
  playTone(5240, 1);
  playTone(5280, 1);
  playTone(5320, 1);
  playTone(5360, 1);
  playTone(5400, 1);
  playTone(5440, 1);
  playTone(5480, 1);
  playTone(5520, 1);
  playTone(5560, 1);
  playTone(5600, 1);
  playTone(5640, 1);
  playTone(5680, 1);
  playTone(5720, 1);
  playTone(5760, 1);
  playTone(5800, 1);
  playTone(5840, 1);
  playTone(5880, 1);
  playTone(5920, 1);
  playTone(5960, 1);
  playTone(6000, 1);
  playTone(6040, 1);
  playTone(6080, 1);
  playTone(6120, 1);
  playTone(6160, 1);
  playTone(6200, 1);
  playTone(6240, 1);
  playTone(6280, 1);
  playTone(6320, 1);
  playTone(6360, 1);
  playTone(6400, 1);
  playTone(6440, 1);
  playTone(6480, 1);
  playTone(6520, 1);
  playTone(6560, 1);
  playTone(6600, 1);
  playTone(6640, 1);
  playTone(6680, 1);
  playTone(6720, 1);
  playTone(6760, 1);
  playTone(6800, 1);
  playTone(6840, 1);
  playTone(6880, 1);
  playTone(6920, 1);
  playTone(6960, 1);
  playTone(7000, 1);
  playTone(7040, 1);
  playTone(7080, 1);
  playTone(7120, 1);
  playTone(7160, 1);
  playTone(7200, 1);
  playTone(7240, 1);
  playTone(7280, 1);
  playTone(7320, 1);
  playTone(7360, 1);
  playTone(7400, 1);
  playTone(7440, 1);
  playTone(7480, 1);
  playTone(7520, 1);
  playTone(7560, 1);
  playTone(7600, 1);
  playTone(7640, 1);
  playTone(7680, 1);
  playTone(7720, 1);
  playTone(7760, 1);
  playTone(7800, 1);
  playTone(7840, 1);
  playTone(7880, 1);
  playTone(7920, 1);
  playTone(7960, 1);
  playTone(8000, 1);
  playTone(8040, 1);
  playTone(8080, 1);
  playTone(8120, 1);
  playTone(8160, 1);
  playTone(8200, 1);
  playTone(8240, 1);
  playTone(8280, 1);
  playTone(8320, 1);
  playTone(8360, 1);
  playTone(8400, 1);
  playTone(8440, 1);
  playTone(8480, 1);
  playTone(8520, 1);
  playTone(8560, 1);
  playTone(8600, 1);
  playTone(8640, 1);
  playTone(8680, 1);
  playTone(8720, 1);
  playTone(8760, 1);
  playTone(8800, 1);
  playTone(8840, 1);
  playTone(8880, 1);
  playTone(8920, 1);
  playTone(8960, 1);
  playTone(9000, 1);
  playTone(9040, 1);
  playTone(9080, 1);
  playTone(9120, 1);
  playTone(9160, 1);
  playTone(9200, 1);
  playTone(9240, 1);
  playTone(9280, 1);
  playTone(9320, 1);
  playTone(9360, 1);
  playTone(9400, 1);
  playTone(9440, 1);
  playTone(9480, 1);
  playTone(9520, 1);
  playTone(9560, 1);
  playTone(9600, 1);
  playTone(9640, 1);
  playTone(9680, 1);
  playTone(9720, 1);
  playTone(9760, 1);
  playTone(9800, 1);
  playTone(9840, 1);
  playTone(9880, 1);
  playTone(9920, 1);
  playTone(9960, 1);
  playTone(10000, 1);
}
```



```
ずっと
アイコンを表示
音を鳴らす 高さ (Hz) 真ん中のド 長さ 1 拍
音を鳴らす 高さ (Hz) 真ん中のミ 長さ 1 拍
音を鳴らす 高さ (Hz) 真ん中のソ 長さ 1 拍
音を鳴らす 高さ (Hz) 上のド 長さ 1 拍
音を鳴らす 高さ (Hz) 上のミ 長さ 1 拍
音を鳴らす 高さ (Hz) 真ん中のソ 長さ 1 拍
音を鳴らす 高さ (Hz) 上のド 長さ 1 拍
音を鳴らす 高さ (Hz) 上のミ 長さ 1 拍
音を鳴らす 高さ (Hz) 真ん中のド 長さ 1 拍
音を鳴らす 高さ (Hz) 真ん中のミ 長さ 1 拍
音を鳴らす 高さ (Hz) 真ん中のソ 長さ 1 拍
音を鳴らす 高さ (Hz) 上のド 長さ 1 拍
音を鳴らす 高さ (Hz) 上のミ 長さ 1 拍
音を鳴らす 高さ (Hz) 真ん中のソ 長さ 1 拍
音を鳴らす 高さ (Hz) 上のド 長さ 1 拍
音を鳴らす 高さ (Hz) 上のミ 長さ 1 拍
音を鳴らす 高さ (Hz) 真ん中のソ 長さ 1 拍
音を鳴らす 高さ (Hz) 下のソ 長さ 1 拍
音を鳴らす 高さ (Hz) 真ん中のミ 長さ 1 拍
音を鳴らす 高さ (Hz) 真ん中のソ 長さ 1 拍
音を鳴らす 高さ (Hz) 上のミ 長さ 1 拍
音を鳴らす 高さ (Hz) 上のファ 長さ 1 拍
音を鳴らす 高さ (Hz) 真ん中のソ 長さ 1 拍
音を鳴らす 高さ (Hz) 上のミ 長さ 1 拍
音を鳴らす 高さ (Hz) 上のファ 長さ 1 拍
音を鳴らす 高さ (Hz) 下のソ 長さ 1 拍
音を鳴らす 高さ (Hz) 真ん中のミ 長さ 1 拍
音を鳴らす 高さ (Hz) 真ん中のソ 長さ 1 拍
音を鳴らす 高さ (Hz) 上のミ 長さ 1 拍
音を鳴らす 高さ (Hz) 上のファ 長さ 1 拍
音を鳴らす 高さ (Hz) 真ん中のソ 長さ 1 拍
音を鳴らす 高さ (Hz) 上のミ 長さ 1 拍
音を鳴らす 高さ (Hz) 上のファ 長さ 1 拍
音を鳴らす 高さ (Hz) 真ん中のミ 長さ 1 拍
音を鳴らす 高さ (Hz) 真ん中のソ 長さ 1 拍
音を鳴らす 高さ (Hz) 上のミ 長さ 1 拍
音を鳴らす 高さ (Hz) 上のファ 長さ 1 拍
```

チャレンジ③のプログラム例  
メロディーを作る

ダウンロード music-sample1

# わくわくプログラミング④ 無線に挑戦

## 目標

最後に2台のmicro:bitで無線通信じゃんけんゲームを作ろう

## やりたいこと

2台のmicro:bitのボタンBを同時に押して無線通信でじゃんけんをし、勝ち、負け、あいこの判定をする。ボタンAで勝ち数を表示させる

たくさんの論理分岐や無線通信などを使ってだいぶ高度になるよ

## 無線通信の手始めに

- ① ボタンを押したら相手のmicro:bitに自分の名前文字や数字を表示させてみよう
- ② 数字や文字でなく、自分のアイコンを表示させるにはどうすればいいのかな？

## プログラミング・ブロックで考えると

最初だけ

無線のグループを設定

1

①無線通信をするmicro:bitのグループを作る（最初に1回だけ）

ボタン A ▼ が押されたとき

無線で文字列を送信

名前

②無線でデータを数字や文字を送信する  
ここでは、「ボタンが押されたとき」イベントブロックを使っている

無線で受信したとき

name ▼

文字列を表示

name ▼

③「無線で受信したときの」イベントブロックを使って、受信した数字や文字を表示させる

それでは、エディタ画面でプログラミングしてみよう！！

## 無線サンプルプログラム 1

最初だけ

無線のグループを設定 0

ボタン A ▼ が押されたとき

無線で文字列を送信 " TARO "

無線で受信したとき receivedString

文字列を表示 receivedString

数を表示 受信したパケットの 信号強度 ▼

receivedString

文字列を受信する無線ブロック専用の変数

receivedNumber

数字を受信する無線ブロック専用の変数

最初だけ

無線のグループを設定 1

## 無線サンプルプログラム 2

ボタン A ▼ が押されたとき

無線で数値を送信 0

ボタン B ▼ が押されたとき

無線で数値を送信 1

無線で受信したとき receivedNumber

もし receivedNumber = 0 なら

アイコンを表示



でなければもし receivedNumber = 1 なら

アイコンを表示



## ゲーム作成にチャレンジ！！

- ①サイコロ対戦ゲームを作ってみよう
- ②「ジャンケン」ゲームを作ってみよう

## 無線サンプルプログラム 3 (サイコロ対戦ゲーム)

最初だけ

無線のグループを設定 1

ボタン A が押されたとき

変数 saikoro を 1 から 6 までの乱数 にする

数を表示 saikoro

無線で数値を送信 saikoro

一時停止 (ミリ秒) 100

無線で受信したとき receivedNumber

もし receivedNumber < saikoro なら

アイコンを表示

でなければもし receivedNumber = saikoro なら

アイコンを表示

でなければ

アイコンを表示



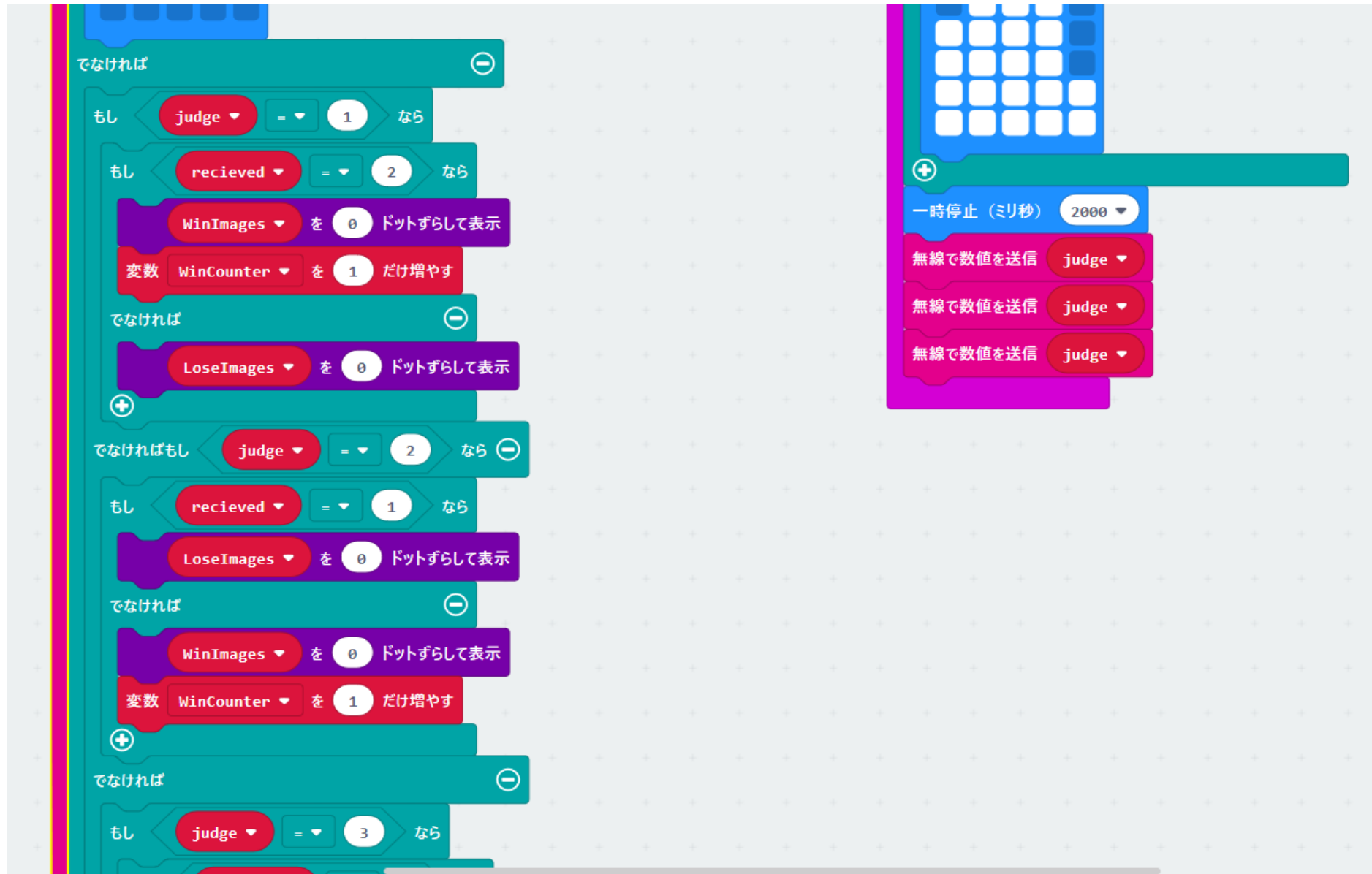
## 無線サンプルプログラム 4 (ジャンケン ゲーム)

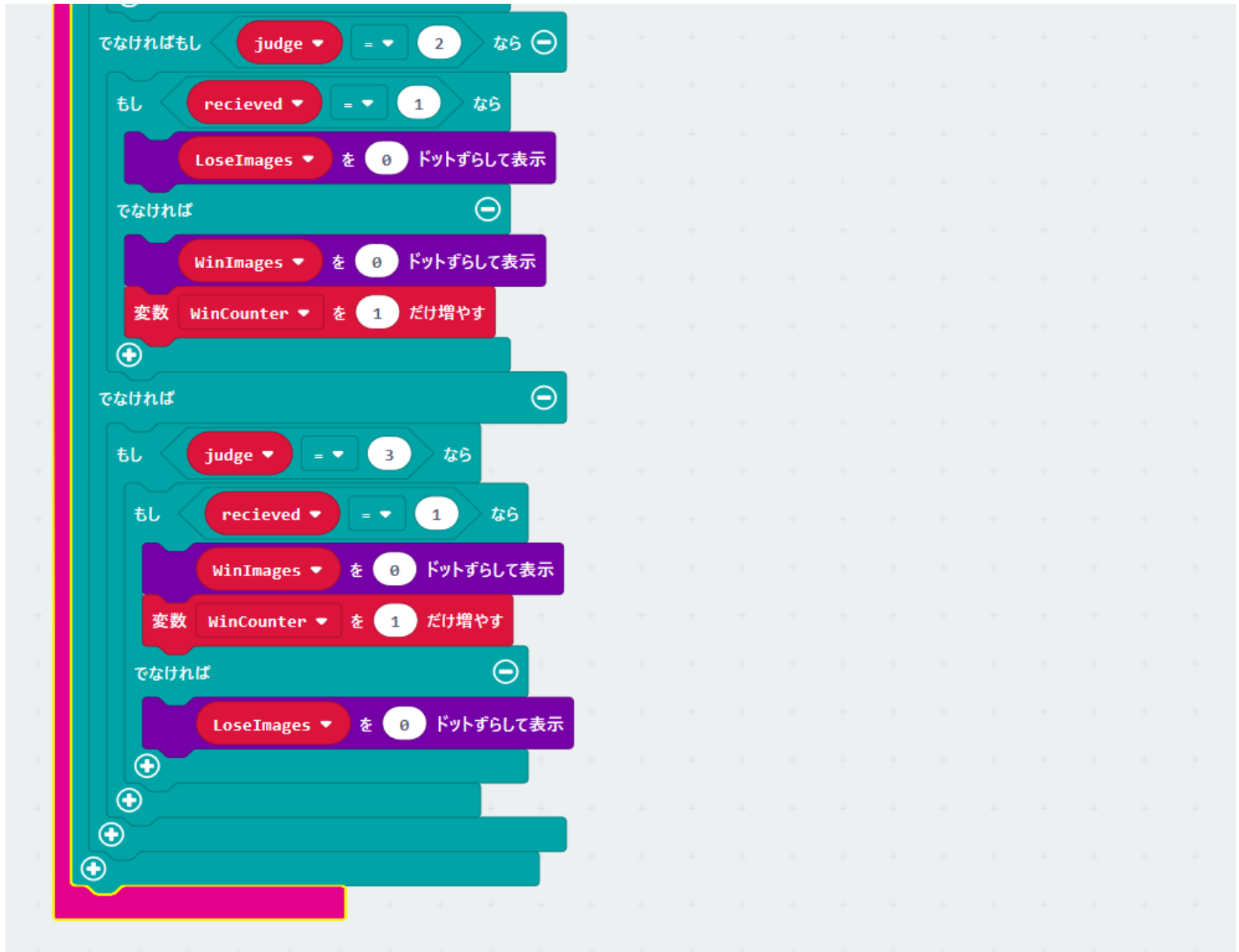
[https://makecode.microbit.org/\\_KeeFpjTU1eVX](https://makecode.microbit.org/_KeeFpjTU1eVX)

The code is organized into three main sections:

- 最初だけ (Only at the start):** This section initializes the game. It sets the radio group to 1, displays the icon, and initializes four variables: WinCounter (0), judge (0), recieved (0), LoseImages (Scissors icon), and WinImages (Rock icon).
- 無線で受信したとき (When received via radio):** This section handles incoming data. It checks if the received value (recieved) is equal to the judge value. If so, it displays the LED screen with the WinImages icon. If not, it checks if judge is 1 or recieved is 2, which would result in a win for the other player.
- ボタン A が押されたとき (When button A is pressed):** This section increments the WinCounter by 1.
- ボタン B が押されたとき (When button B is pressed):** This section generates a random number (judge) between 1 and 3, displays the LED screen with the judge value, and checks if judge is 1 or 2 to determine the outcome.







サポートしてくれた鞍手高校のお兄さんたち  
から、プログラム作品のプレゼント！！

(PIN-PON GAME)のプレゼント

[https://makecode.microbit.org/\\_eH6Fe5eou21q](https://makecode.microbit.org/_eH6Fe5eou21q)

OmakeCodeエディタのホーム画面右 **読み込む** からURLを読み込んでみてください

# 最後に、自分でいろいろ調べ、何か作って、 第3回目でみんなに見せてみよう！！

(プログラミングの参考ホームページ)

micro:BITのオリジナルホームページ <https://microbit.org/ja/>  
サヌキテックネット <https://sanuki-tech.net/micro-bit/>

(ネット通販で関連部品を購入するには)

スイッチサイエンス <https://www.switch-science.com/>  
Amazonやモノタロウ などでも部品が手に入る

(いろんな質問、疑問があったら)

<https://adox.jp/kids-programming/>  
の「プログラミング教室についてのお問い合わせから」メールで聞いてください。

# 次世代 キッズプログラミング教室 第3回目

第1回目：2020年1月26日

micro:BITボードについての基礎、基本表示、センサーの利用

第2回目：2020年2月2日

外部出力、無線通信、おまけ

第3回目：2020年2月24日

総合演習（復習）みんなが作ったものの発表会など

# これまでにやったこと

## ○第1回目：2020年1月26日

micro:BITボードの機能などについて、LED画面の基本表示、内蔵センサー等などの使い方などについてやりました。

## ○第2回目：2020年2月2日

外部出力端子を使ってLEDライトを作ったり、スピーカから音を出したり。2台のmicro:BITの間で、無線通信機能を使ったジャンケンゲーム、登録してあるプログラムの取りこみをやりました。

## ○第3回目：2020年2月24日

総合学習（復習）、みんなが作ったものを見せ合う、遊ぶ

## 今日、やること

- ①プログラミングにとって大事なことを覚えておこう
- ②何か新しく作ったものを見せてみよう
- ③鞍手高校のお兄さん、お姉さんたちが作ったもので遊んでみよう

# (総合復習) プログラミングにとって大事なことを覚えておこう

1. まず、コンピュータに何をさせるかの目当てを決める  
温度計を作りたい、こんなゲームのアプリを作りたい . . . . .
2. コンピュータにやらせることの手順を順序だてて、細かく分解する  
入力はどうするか、キーボード?、マウス?、タッチ? . .  
この時はこの画面を出力させる、こんな動きをさせる . . .
3. 手順のつながりを考える (フローチャート)  
このデータなら、こちらを実行する。違うならこちらに進む . . .
4. 適合するプログラミング言語を選ぶ  
MakeCode、Scratch、Python、Java、JavaScript、Swift、C++ . . . . .
5. エディタを使ってプログラミングする (コーディング)
6. 思いどおりに動いたか確かめ、悪い部分のプログラムを修正していく (デバグ)



Scratch
ファイル
編集
チュートリアル
Scratchに参加しよう
サインイン

コード
コスチューム
音

動き
動き
10 歩動かす
見た目
15 度回す
音
15 度回す
イベント
どこかの場所へ行く
制御
x座標を0, y座標を0にする
関わる
1秒でどこかの場所へ行く
演算
1秒でx座標を0に, y座標を0に
変数
90度に向ける
ブロック定義
マウスのポインターへ向ける
x座標を10ずつ変える
x座標を0にする

スプライト
スプライト1
表示する
向き
90
サイズ
100

ステージ
背景
1

```

66
67
68
69
70
71
72 let WinImages: Image = null
73 let LoseImages: Image = null
74 let judge = 0
75 let WinCounter = 0
76 radio.setGroup(1)
77 basic.showIcon(IconNames.Heart)
78 WinCounter = 0
79 judge = 0
80 let recieved = 0
81 LoseImages = images.iconImage(IconNames.Sad)

```

## (付録) コンピュータってどうなってるの？

### ※コンピュータは、最終的には1か0しか理解できない

・コンピュータは1,0の並びで全てのことを計算、判断していて、1,0の並びを一度にどれだけ計算できるかで性能が決まる。(4bit、8bit、16bit、32bit、64bit・・・)

### ※人間の算数の世界では、10進数（デシマル）を使って計算するが、コンピュータの世界では16進数、8進数、2進数を使う

◆2進数（バイナリ）：00=0、01=1、10=2、11=3、100=4、101=5、110=6、111=7、1000=8、1001=9、1010=10、1011=11、1100=12、1101=13、1110=14、1111=15

◆8進数（オクタル）：00、01、02、03、04、05、06、07、10=8、11=9、12=10・・・77=255

◆16進数（ヘキサデシマル）：00、01、02、03、04、05、06、07、08、09、0A、0B、0C、0D、0E、0F=15・・・FF=32767

○MakeCodeでは、書かれたブロックプログラムを16進数に変換し（～～.hex）micro:BITに渡す。

○micro.BITは渡された16進数を順番どおりに2進数に置き換えながら実行していく。

```
microbit-game-janken.hex - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)

:0200000040000FA
:10000000C0070000D1060000D1000000B1060000CA
:1000100000000000000000000000000000000000E0
:1000200000000000000000000000000000000005107000078
:1000300000000000000000000000000000000000DB000000E500000000
:10004000EF000000F900000003010000D010000B6
:1000500017010000210100002B0100003501000004
:100060003F01000049010000530100005D01000054
:1000700067010000710100007B01000085010000A4
:100080008F01000099010000A3010000AD010000F4
:10009000B7010000C1010000CB010000D501000044
:1000A000DF010000E9010000F3010000FD01000094
:1000B00007020000110200001B02000025020000E0
:1000C0001FB5C046C04600FoEFA04BooFB41FBD24
:1000D00008205A49096809580847382057490968CB
:1000E000095808473C2055490968095808474020E5
:1000F0005249096809580847442050490968095875
:10010000084748204D490968095808474C204B4981
:10011000096809580847502048490968095808479C
:100120005420464909680958084758204349096836
:10013000095808475C204149096809580847602068
:100140003E4909680958084764203C49096809582C
:100150000847682039490968095808476C20374919
:100160000968095808477020344909680958084740
:100170007420324909680958084778202F490968CE
:10018000095808477C202D490968095808478020EC
:1001900002A490968095808478420284909680958E4
:1001A0000847882025490968095808478C202349B1
:1001B00009680958084790202049096809580847E4
:1001C00094201E4909680958084798201B49096866
:1001D000095808479C201949096809580847A02070
:1001E0001649096809580847A201449096809589C
:1001F0000847A8201149096809580847AC200F4949
:10020000096809580847B0200C4909680958084787
:10021000B4200A49096809580847B82007490968FD
:1002200009580847BC20054909680958084709000D3
:1002300003480449024A034B7047000000000020B5
:10024000C0070000C00700000122D84B5A6000BF61
:10025000D74A1268002AFBD0016000BFD44A126856
:10026000002AFBD00022D14B5A6000BFD04A12684E
```

microbit

PC > ボリューム (D:) > Fujimoto2018.4 > microbit

名前	更新日時	種類	サイズ
bitBot	2020/02/13 14...	ファイル フォルダー	
microbit-BT_test1.hex	2019/01/31 13...	Microsoft Mak...	615 KB
microbit-fibonacci.hex	2019/02/04 14...	Microsoft Mak...	598 KB
microbit-game-janken.hex	2019/02/13 12...	Microsoft Mak...	611 KB
microbit-Hayaosi-counter.hex	2019/02/07 11...	Microsoft Mak...	605 KB
microbit-KEISAN-KI.hex	2019/01/22 14...	Microsoft Mak...	600 KB
microbit-LED.hex	2020/01/17 9...	Microsoft Mak...	626 KB
microbit-LED2.hex	2019/04/12 15...	Microsoft Mak...	596 KB
microbit-LED3.hex	2019/04/12 15...	Microsoft Mak...	595 KB
microbit-LOVE-METER.hex	2019/01/25 16...	Microsoft Mak...	604 KB
microbit-magic_button.hex	2019/01/31 13...	Microsoft Mak...	597 KB
microbit-moter-reciever.hex	2019/06/11 12...	Microsoft Mak...	608 KB
microbit-motor-controla.hex	2019/05/24 11...	Microsoft Mak...	606 KB
microbit-motor-controla-3.hex	2019/06/11 13...	Microsoft Mak...	607 KB
microbit-musen-sample2.hex	2019/11/22 13...	Microsoft Mak...	632 KB
microbit-music-suple1.hex	2019/05/31 9...	Microsoft Mak...	615 KB
microbit-ping-pong-game.hex	2019/07/17 13...	Microsoft Mak...	629 KB
microbit-radio-send-test1.hex	2019/02/06 13...	Microsoft Mak...	608 KB
microbit-Raumen-Timer.hex	2019/01/28 15...	Microsoft Mak...	610 KB
microbit-rollpich.hex	2019/01/29 16...	Microsoft Mak...	597 KB
microbit-rollpich3.hex	2019/01/30 16...	Microsoft Mak...	601 KB
microbit-Ryusei-Fash.hex	2019/02/14 16...	Microsoft Mak...	606 KB
microbit-sensor-test4.hex	2019/06/26 10...	Microsoft Mak...	641 KB
microbit-tatch-sensor.hex	2019/01/28 15...	Microsoft Mak...	603 KB
microbit-test1.hex	2019/01/25 14...	Microsoft Mak...	608 KB
microbit-test1-1.hex	2019/01/29 16...	Microsoft Mak...	603 KB
microbit-test2.hex	2019/01/25 13...	Microsoft Mak...	608 KB
microbit-test3.hex	2019/01/25 15...	Microsoft Mak...	606 KB

Lesson 10 APP  
control1811...

bit...

年度  
ソフト

data

ku  
a

P1180078.jpg P1180056.jpg P1180065.jpg  
P1180081.jpg P1180071.jpg P1180084.jpg P1180064.jpg