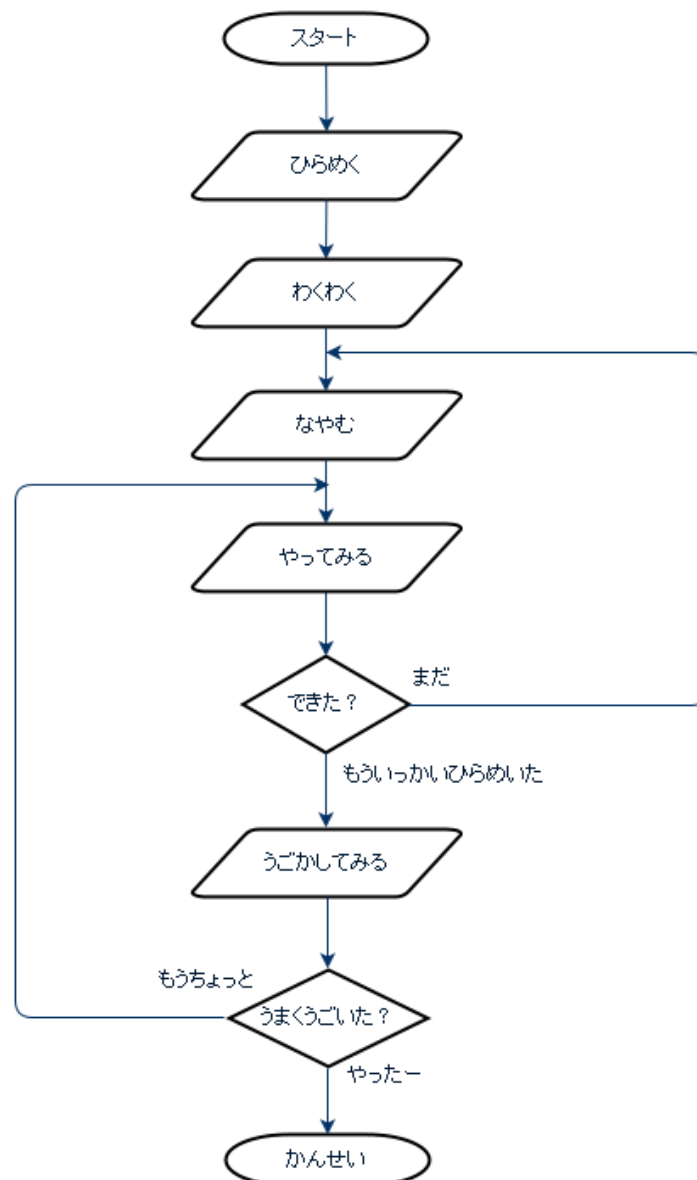


はじめてのプログラミング

Micro:bit バージョン (経験者向け)



■はじめに

①Micro:bit ってなに？

Micro:bit は、イギリスで開発された教育向けマイコンボードです。

それでは**マイコン**とは何でしょう？ マイコンは**小さなコンピューター(マイクロコンピューター)**です。マイコンは、携帯電話やテレビ、冷蔵庫、おもちゃなどに使われているコンピューターで、**機械(きかい)を制御(せいぎよ)する頭脳(ずのう)**になります。

このマイコン **Micro:bit** もプログラムがなければ動きません。Micro:bit にプログラムを送りこんで、機械を制御してみよう。

②Micro:bit のプログラムで制御できるもの

- 25 個の LED(明るさセンサとしても使える)
- 2 個のボタンスイッチ
- 加速度センサー
- 磁力センサー
- 温度センサー(ARM CortexR-M0 プロセッサに内蔵)
- Bluetooth Low Energy モジュール(HRM1017)

③Micro:bit のプログラムを作るには？

- **Java スクリプトブロックエディタ**を使います(日本語対応)
スクラッチに似た操作でブロックを使ってプログラミングができます。

<https://makecode.microbit.org/>

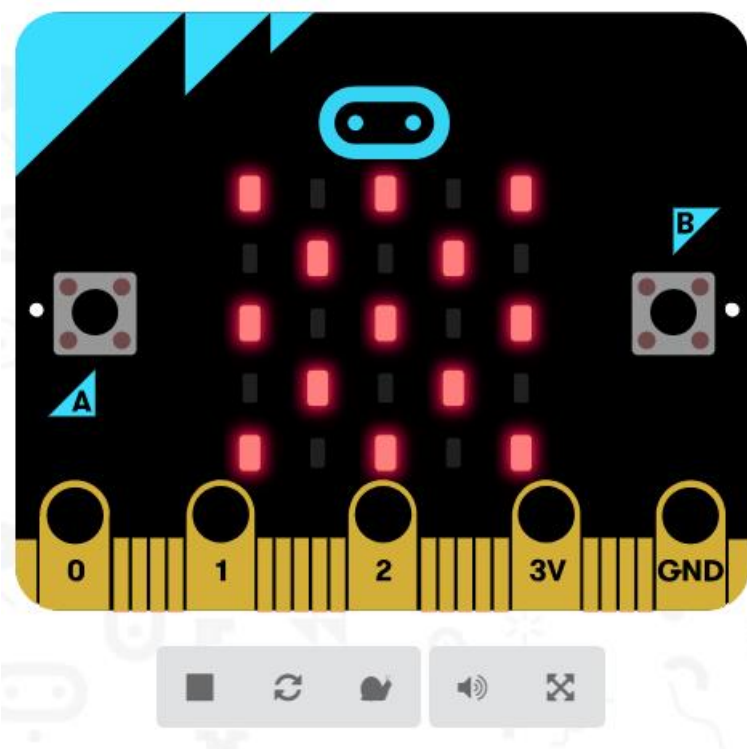
■練習1 マイコンを使ったはじめてのプログラミング

①エディタにアクセスします。(<https://makecode.microbit.org//>)

②下記のプログラムを組もう。



③シミュレータ(左側)で、ちゃんと動くか確認しよう。



④名前を付けて保存しよう。下記では「ren1」の名前を付けて保存してみました。パソコンのどこに保存されたか分からない場合は、先生に聞いてね。



Google Chrome のブラウザを使っている場合は、画面の下に保存したファイルが表示されます。右クリックして「フォルダを開く」を選択しましょう。

ブラウザが Google Chrom の場合



ブラウザが Internet Explorer の場合

ファイルを保存すると画面下側に下記のウィンドウが表示されます。「保存」の▼を押して、「名前を付けて保存」を選びます。



ファイルが保存できると画面下側に下記のウィンドウが表示されます。「フォルダー」

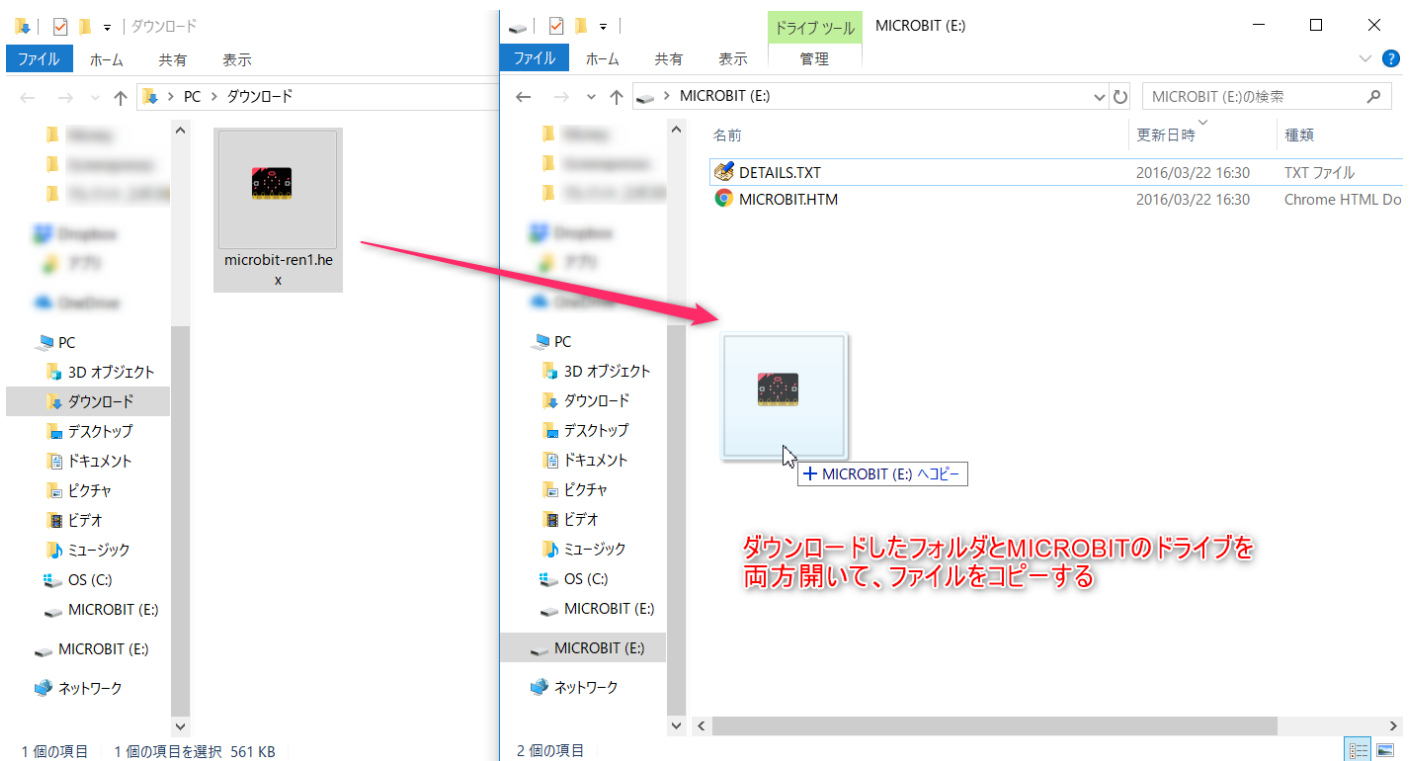


⑤パソコンの USB ポートに、マイコンを接続します。

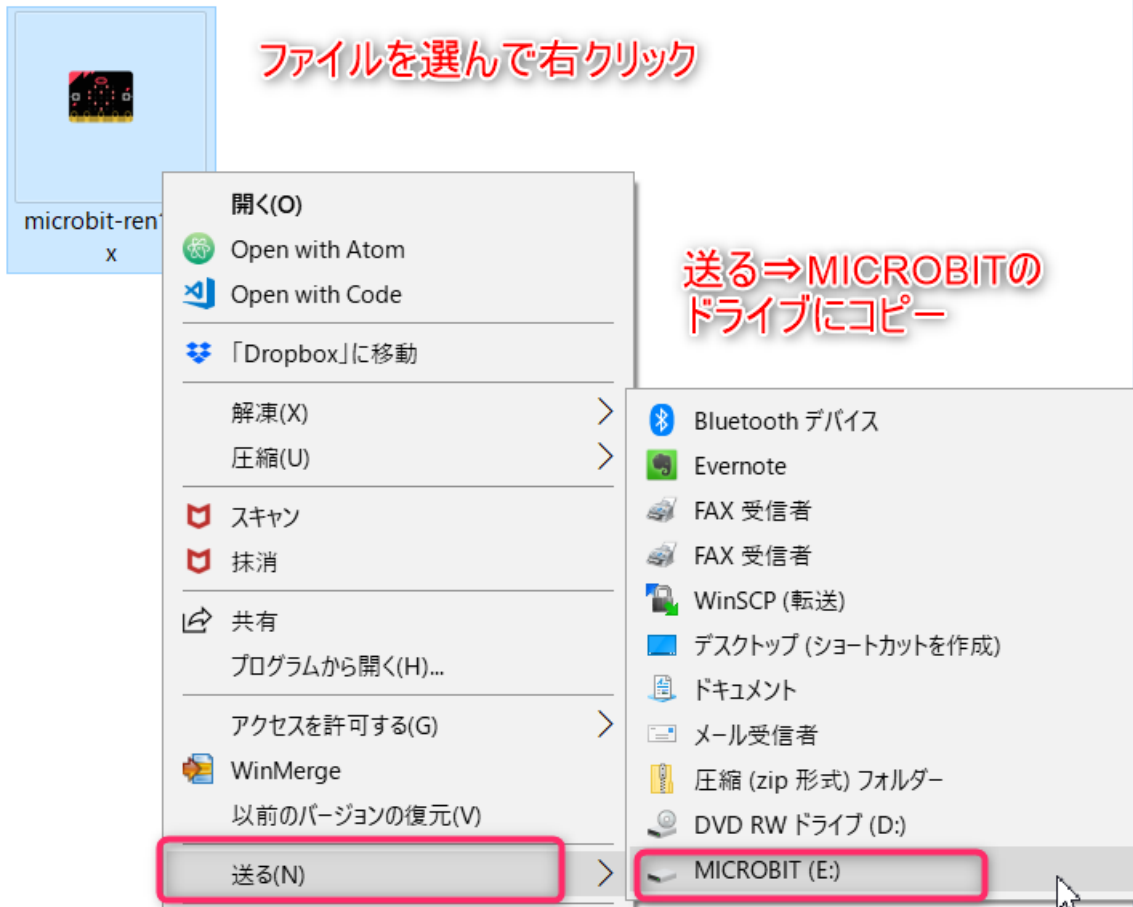
※取扱いに注意してください。

⑥マイコンにプログラムファイルをコピーします。コピーの方法を二通り説明します。

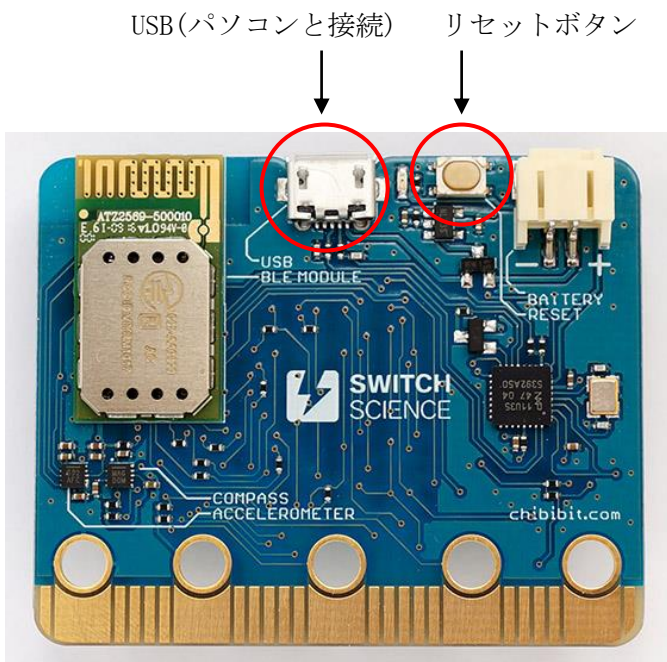
方法 1 :



方法 2 :



⑦マイコンのリセットボタンを押して、プログラムを開始！



⑧マイコンのLEDは光ったかな？ どんな動きをするプログラムか説明してみよう。

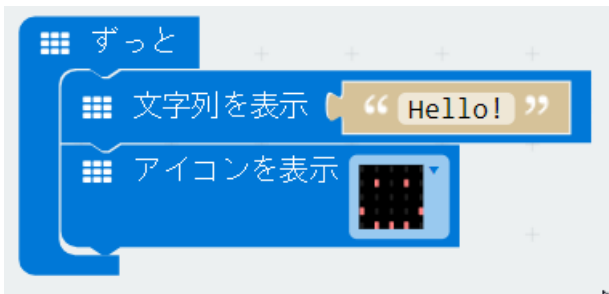
■練習 2 LED に数字を表示させよう

LED に数字 0~9 を順番に表示させよう。“くり返し”と“変数”を使います。



■練習 3 LED に文字やアイコンを表示させよう

LED に“Hello!”とアイコン（うれしい顔）を表示させます。



■練習 4 ボタンを使おう

ボタン A を押したら数字が増えるプログラムを作ろう。



♣かんがえよう 1

ボタン A を押したら数字が増えるプログラムを作ろう。ただし、ボタン B を押したら数字を消して 0 に戻るプログラムにすること。

♣かんがえよう 2

下記ルールのカウントアッププログラムを作ろう。

- ・ルール 1 : ボタン A を押すと変数 i の値が増えていく (まだ数字は表示しません)
- ・ルール 2 : ボタン B を押すと変数 i の現在の値を表示する → 3 秒後に消える
- ・ルール 3 : ボタン A+B を同時に押すと、変数 i を 0 にする

♣かんがえよう 3

乱数で決めた 0~5 の数字をピッタリあてるゲームを作ろう。友達に数字を言い当ててもらおう。

- ・ルール 1 : ボタン A を押すと乱数で 0~5 の数字を変数 i にセットします。
- ・ルール 2 : ボタン B を押すと変数 i の値を表示します。(答え合わせ)

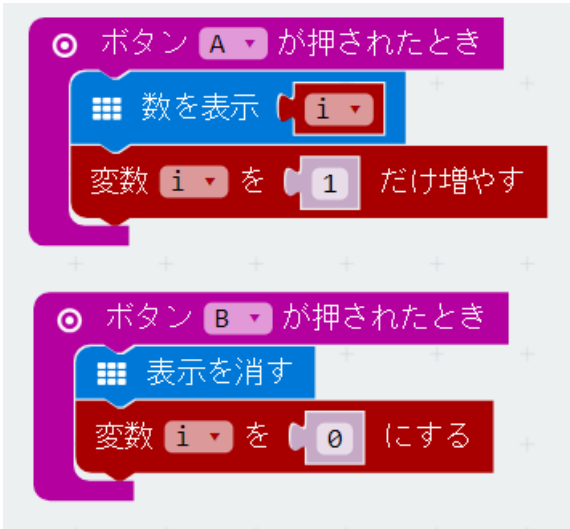
※くり返しゲームができるように、もう一度ボタン A を押すと表示が消えます。

♣かんがえよう 4

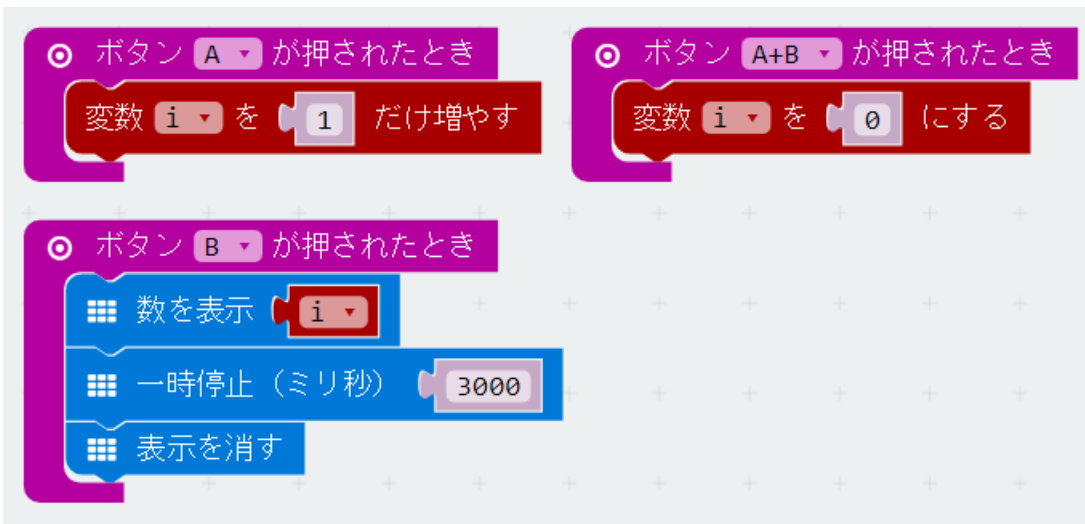
2 つの数字を表示して、足し算をするゲームを作ろう。

- ・ルール 1 : スタートボタンを押すと乱数で決めた 0~9 の数字を 2 個表示します。(2 秒ずつ表示後に消える)
- ・ルール 2 : ボタン A を押して答えをセットします。押した数だけ答えが増えて行きます。
- ・ルール 3 : ボタン B を押して結果を表示します。正解ならレ点のアイコン、間違いなら×のアイコンを表示。

♣ かんがえよう 1 の答え



♣ かんがえよう 2 の答え



♣ かんがえよう 3 の答え

```

    〇 ボタン A が押されたとき
        表示を消す
        変数 i を 0~5 の範囲の乱数 にする

    〇 ボタン B が押されたとき
        数を表示 i
    
```

♣ かんがえよう 4 の答え

```

    最初だけ
        変数 result を 0 にする
        変数 i を 0~9 の範囲の乱数 にする
        数を表示 i
        一時停止 (ミリ秒) 2000
        変数 j を 0~9 の範囲の乱数 にする
        数を表示 j
        一時停止 (ミリ秒) 2000
        表示を消す

    〇 ボタン A が押されたとき
        変数 result を 1 だけ増やす

    〇 ボタン B が押されたとき
        もし (result = i + j)
            アイコンを表示 [LED 点滅]
        でなければ
            アイコンを表示 [LED 点滅]
    
```

■練習 5 加速度センサーを使おう

マイコンを揺らすと数字が増えるプログラムを作ろう

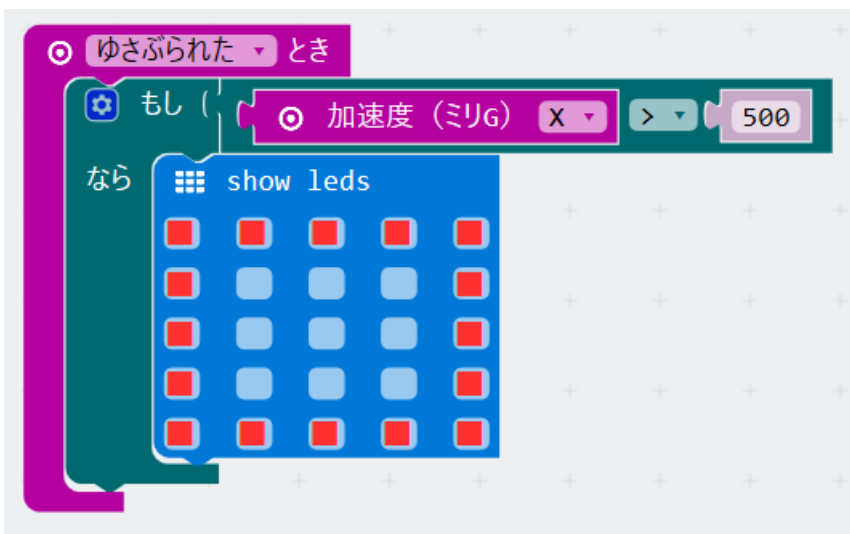
※ゆらすときは落とさないように注意してね！



■練習 6 加速度センサーを使おう

X 方向にマイコンをゆらして、加速度が 500 ミリ G 以上の時に LED を表示させよう。

※ゆらすときは落とさないように注意してね！



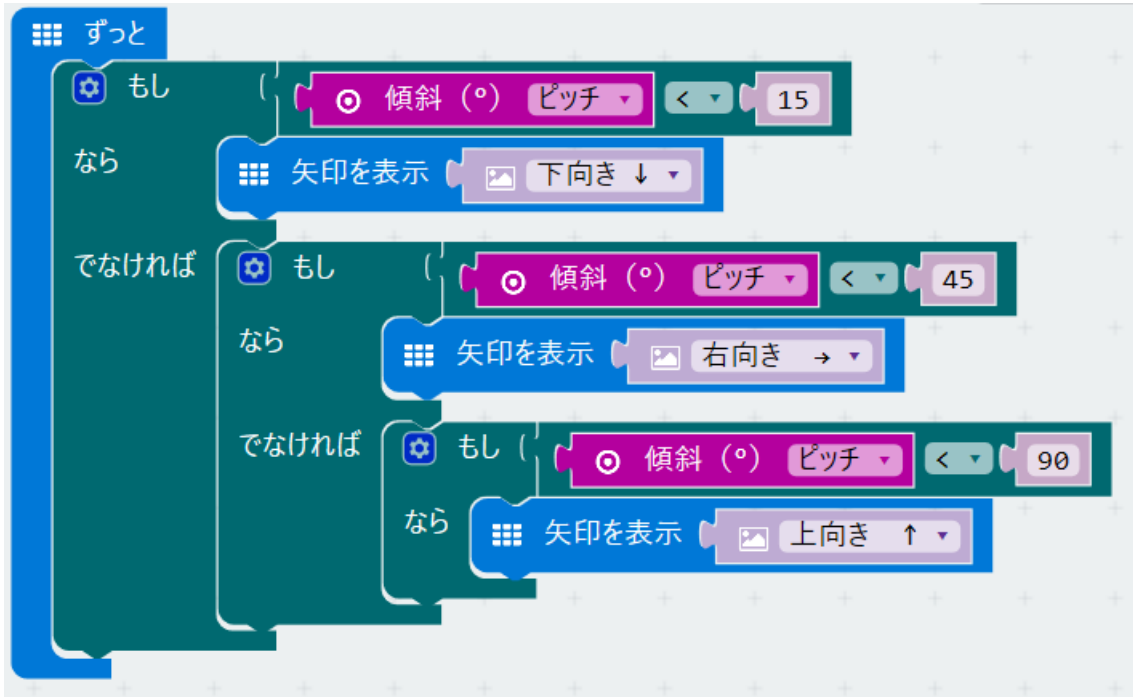
■練習 7 温度センサーを使おう

マイコンのマイコン温度を表示させよう。(マイコン温度は部屋の温度より高めです)



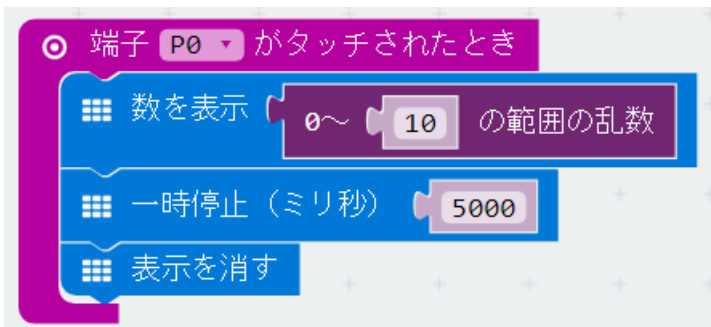
■練習 8 傾きセンサーを使おう

LED に矢印を表示させます。マイコンの傾きを検出して矢印の向きを変えるプログラムを作ります。マイコンの向きを変えて、矢印が変わるかを確認しよう。



■練習 9 端子を使おう

端子を使った相性占いプログラミングを作ります。一人が端子 0 をもう一人が GND をタッチして、数字が大きいほど相性が良い占いとします。試してみてください。

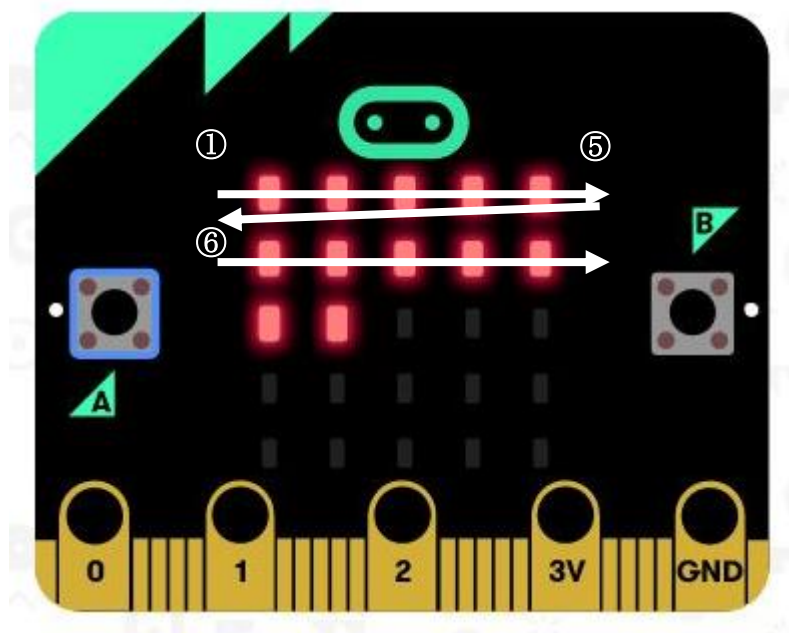


■かんがえよう 7 押ただけ光る LED

ボタン A を押した数だけ LED が光るプログラムを作ろう。

ボタン A を 1~5 回目まで押したときは、上から一列目の左から順に点灯していく。

6~10 回目まで押したときは上から二列目の左から順に点灯していくよ。



□かんがえよう 7 解答例

The image shows a sequence of code blocks for a Micro:bit program, organized into four sections:

- 最初だけ (Initially):** A blue block containing three red blocks: "表示を消す" (Clear display), "変数 横 を 0 にする" (Set variable 横 to 0), and "変数 縦 を 0 にする" (Set variable 縦 to 0).
- ずっと (Forever):** A blue block containing a green loop block with three sub-blocks: "変数 j を 0 から 縦 に変えてくりかえす" (Repeat with variable j from 0 to 縦), "変数 i を 0 から 横 に変えてくりかえす" (Repeat with variable i from 0 to 横), and a purple block "点灯 x i y" (Turn on LED at x i y).
- ボタン A が押されたとき (When button A is pressed):** A purple block containing: "変数 横 を 1 だけ増やす" (Increase variable 横 by 1), a teal "もし (横 > 4) なら" (If 横 > 4) block, and two red blocks: "変数 縦 を 1 だけ増やす" (Increase variable 縦 by 1) and "変数 横 を 0 にする" (Set variable 横 to 0).
- ボタン B が押されたとき (When button B is pressed):** A purple block containing three red blocks: "表示を消す" (Clear display), "変数 横 を 0 にする" (Set variable 横 to 0), and "変数 縦 を 0 にする" (Set variable 縦 to 0).