Ⅰ. micro:bit のプログラム

Ⅰ-1　「最初だけ」ブロックの利用（ハートの表示）（プログラム prei1-1）

「最初だけ」ブロック

＊「基本」-「LED 画面に表示」

Ⅰ-2　「ずっと」ブロックの利用（ハートの点滅）（プログラム prei1-2）

「ずっと」ブロック

＊「基本」-「一時停止」「表示を消す」

２．プログラムの基本（順次構造、反復構造）

2-1　順次構造（プログラム prei2-1）

「ずっと」ブロック

＊「基本」-「アイコン表示」（ハート）

＊「基本」-「一時停止」

＊「基本」-「表示を消す」

2-2　反復構造（繰り返し 5 回）（プログラム prei2-2）

「最初だけ」ブロック

＊「基本」-「アイコン表示」の「はさみ」を選択

　じゃんけんの「グー」「チョキ」「パー」表示とする。

＊「ループ」－「くりかえし」　数値の 0→ 5

反復構造は、繰り返し構造ともいう。

別解（カウンター利用）（繰り返し５回）（プログラム prei2-3）

「最初だけ」ブロック

＊「ループ」の「変数 カウンター ～ くりかえす」 数値の 0→ 4

変数「カウンター」は、0 から始まり、「0，1，2，3，4」の５回、繰り返す

３．プログラムの基本（分岐構造）

3-1 分岐構造（分岐２つ）（プログラム prei3-1）

「最初だけ」ブロック

＊「変数」-「変数を追加する」 → 「c」にする

＊「変数」-「変数 c を 0 にする」

＊「論理」-「条件判断」（もし なら～ でなければ～）

＊「論理」-「くらべる」-「0 ＝▽ 0」 → 「c ＝▽ 0」 にする

分岐構造は、選択構造ともいう。

順次構造、反復構造(繰り返し構造)、分岐構造(選択構造)をプログラムの基本構造という。

3-2　分岐構造（分岐２つ、乱数の利用）（プログラム prei3-2）

「ずっと」ブロック

＊「変数」-「変数を追加する」　→　「c」にする

＊「計算」-「0～10 までの乱数」　数値の 10→　1（乱数は 0、1 のいずれか）

＜参考＞　micro:bit の Python への自動変換プログラム

```
(1) 反復構造（prei2－3）
  for カウンター in range(5):
      basic.show_icon(IconNames.SMALL_DIAMOND)
      basic.pause(500)
      basic.show_icon(IconNames.SCISSORS)
      basic.pause(500)
      basic.show_icon(IconNames.SQUARE)
      basic.clear_screen()

(2) 分岐構造（prei3－1）
  c = 0
  if c == 0:
      basic.show_icon(IconNames.SMALL_DIAMOND)
  else:
      basic.show_icon(IconNames.SQUARE)
```

４．コンピュータとじゃんけん

4-1　アイコン（グー、チョキ、パー）表示（分岐３つ）（プログラム prei4-1）

「ずっと」ブロック

＊「変数」－「変数を追加する」　→　「c」にする

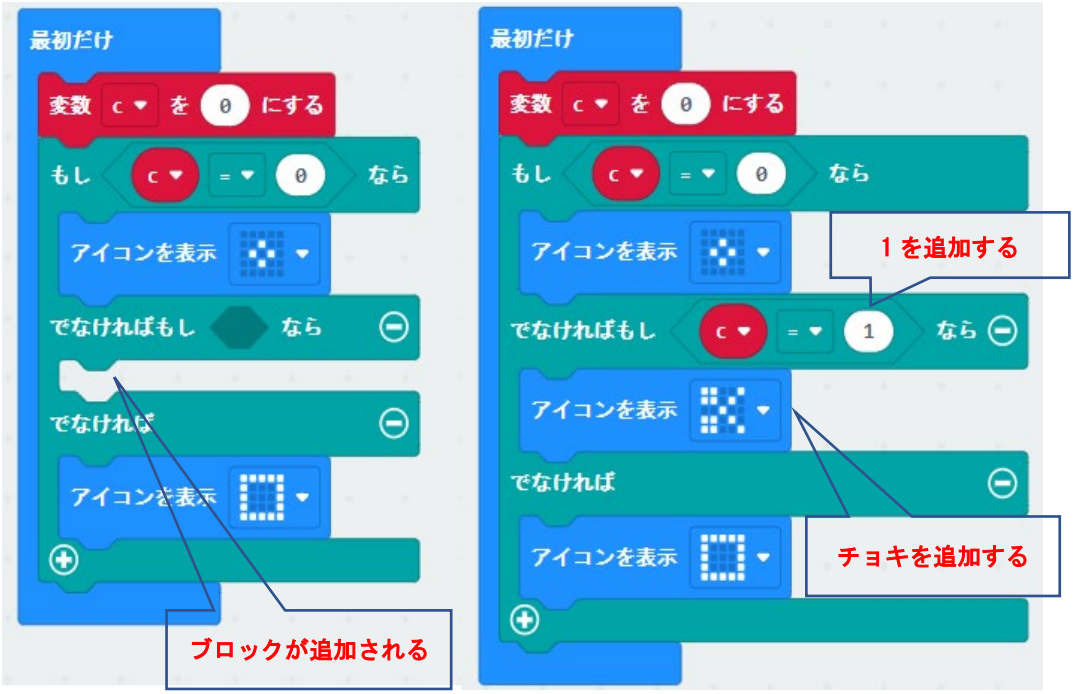
＊「変数」－「変数ｃを０にする」

＊「論理」－「条件判断」（もし　なら～　でなければ～）で、「⊕」をクリックする

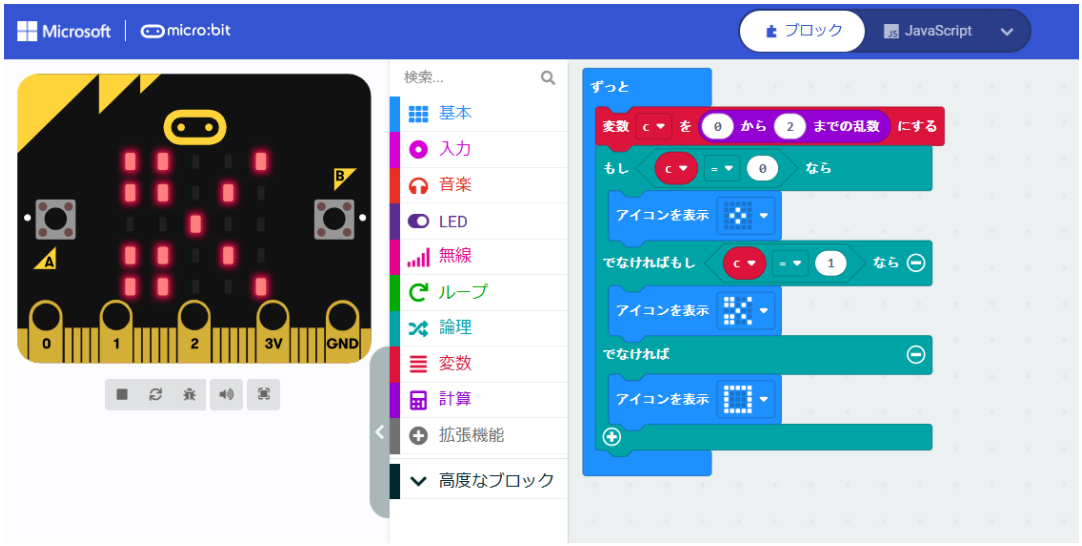
（もし　なら～　でなければもし　なら～　でなければ～）のブロックになる

＊「でなければもし　なら」の箇所、「ｃ　＝▼　１」　を追加する

＊「基本」－「アイコン表示」で、「チョキ」を追加する



4-2　アイコン（グー、チョキ、パー）表示（乱数の利用）（プログラム prei4-2）



「ずっと」ブロック

＊「変数」-「変数を追加する」　→　「c」にする

＊「計算」-「0～10 までの乱数」　数値の 10→　2（乱数は 0、1、2 のいずれか）

4-3　コンピュータの手（繰り返しは 5 回）（プログラム prei4-3）

「最初だけ」ブロック

＊「基本」－「アイコン表示」で、「グー」を表示する

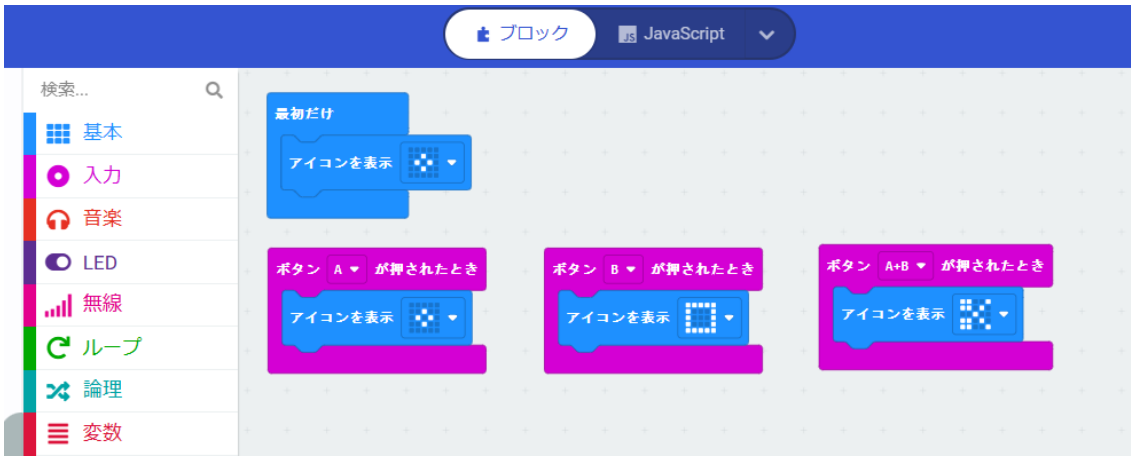
＊「カウンター」（0～4）で、5 回繰り返す

＊「基本」－「数を表示」で、カウンターの数値（回数）を表示する

＊自分の番の時は、「矢印を表示」で「右向き」（→）を表示して、2 秒待つ

＊コンピュータの時は、「文字列を表示」で「Com」と表示する

4-4　自分の手（スイッチボタンを利用）（プログラム prei4-4）



「最初だけ」ブロック

＊「基本」－「アイコンを表示」で「グー」を表示する

＊「入力」－「ボタン A が押されたとき」を選択する

＊A ボタンでは「グー」、B ボタンでは「パー」、A+B ボタンでは「チョキ」とする。

＜参考＞　勝敗の判定表

じゃんけんの勝敗を判定するプログラムを考えてみよう。

A さん、B さんの 2 人の勝敗を判定する表は、下の通りです。まずは、この表の「判定」欄から右列の欄の意味を考えてみよう。

種類	数値	A	B	判定	（A－B）の値	（A-B+3）/3 の余り
グー	0	0	0	引分け	0	0
		0	1	A	-1	2
		0	2	B	-2	1
チョキ	1	1	0	B	1	1
		1	1	引分け	0	0
		1	2	A	-1	2
パー	2	2	0	A	2	2
		2	1	B	1	1
		2	2	引分け	0	0

＜参考文献＞

高橋、喜家村、稲川：micro:bit で学ぶプログラミング、pp.14-15、pp.41-43、コロナ社(2019)

＜参考＞「グー」「チョキ」「パー」の出す回数（プログラム prei4-5）

コンピュータの手を 100 回実施したとき、「グー」「チョキ」「パー」の出す回数(実際は、乱数の出す数値(0,1,2)の回数)を調べるプログラムを考えてみよう。下のプログラム例は、「ブロック」から「JavaScript」欄の「Python」を選択し、表示したものです。


```
1  c = 0
2  k0 = 0
3  k1 = 0
4  k2 = 0
5  for カウンター in range(100):
6      c = randint(0, 2)
7      if c == 0:
8          k0 += 1
9      elif c == 1:
10         k1 += 1
11     else:
12         k2 += 1
13 basic.show_string("G=")
14 basic.show_number(k0)
15 basic.clear_screen()
16 basic.show_string("C=")
17 basic.show_number(k1)
18 basic.clear_screen()
19 basic.show_string("P=")
20 basic.show_number(k2)
21 basic.clear_screen()
```

このプログラムを作成してみよう。

新しいプロジェクト(例えば、「sankou」とする)を作成し、「JavaScript」欄から「Python」を選択し、まずは、このプログラムの 12 行目までを入力し、エラーがでれば修正し、エラーがでなくれば、ブロックで表示してみよう。

つぎに、見やすい結果の表示方法を考えてみよう。右は、先のプログラムの「ブロック」での「回数」を表示した例です。