小中学生のための micro:bit を利用した

プログラミング教室（第３回）資料

氏　名＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿

主催　NPO 法人 学習開発研究所

後援　京田辺市教育委員会

講習(１)：コンピュータとじゃんけんしてみよう（テーマ４）（基礎）

日　時：2025 年９月 21 日(日)　10:00～12:30

講習(２)：コンピュータとじゃんけんしてみよう（テーマ４）（応用）

日　時：2025 年９月 21 日(日)　13:30～16:00

場　所：京田辺市中央公民館　２F　第３・４研修室

問い合わせ先：ild-kemsyu@u-manabi.org
担当者：NPO 法人 学習開発研究所　三輪、高橋

テーマ４　コンピュータとじゃんけんしてみよう

目　次

注）例題番号は、「小中学生のためのプログラミング～学習ガイド」の例題番号に対応しています。
本文中の例題の右端の、「☆」は応用、「★」は発展、「無印」で基礎の問題を示しています。

注）基礎は、主に1.～4.を実施し、応用は、4.と5.を実施します。

＜小中学生のためのプログラミング教室＞

https://u-manabi.net/ild-pkouza/

＜参考文献＞

高橋参吉、喜家村奨、稲川孝司：micro：bit で学ぶプログラミング、ブロック型から JavaScript そして Python へ、コロナ社(2019)

https://u-manabi.net/microbit/

1．micro:bit のプログラム

【例題 1-1】 ハートの表示（プログラム prei1-1）

「最初だけ」ブロックを利用して、ハートを表示してみよう。

「最初だけ」ブロック

＊「基本」-「LED 画面に表示」 LED をタップして、ハート形に見えるように LED を ON にする。

【例題 1-2】 ハートの点滅（プログラム prei1-2）

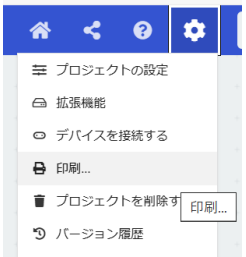
「ずっと」ブロックに変更して、ハートを点滅してみよう。

「ずっと」ブロック

＊「基本」-「一時停止」 時間を 500（ミリ秒）にする。

＊「表示を消す」

＊「基本」-「一時停止」 時間を 500（ミリ秒）にする。



＜参考＞ プログラムの印刷

エディタの画面(右上端)にある歯車（⚙）から「印刷」を選択する。少し経過して「プログラムを印刷する」の画面が表示され、ブロックのプログラムを印刷できる。

2．プログラムの基本（順次構造、反復構造）

【例題 2-1】 アイコンの表示（プログラム prei2-1）

3つのアイコン（「ハート」「小さいダイアモンド」「しかく」）を表示してみよう。

「最初だけ」ブロック

＊「基本」-「アイコン表示」(ハート)、「基本」-「一時停止」

＊「基本」-「アイコン表示」(小さいダイアモンド)、「基本」-「一時停止」

＊「基本」-「アイコン表示」(しかく)、「基本」-「一時停止」

＊「基本」-「表示を消す」

【例題 2-2】 アイコンの繰り返し表示 （プログラム prei2-2）

アイコンを 3 つ（「小さいダイアモンド」「はさみ」「しかく」）を繰り返し 5 回表示してみよう。

「最初だけ」ブロック

＊「基本」-「アイコン表示」の「はさみ」を選択する。

＊「ループ」-「くりかえし」 数値 0 を 5 にする。

じゃんけんの「グー」「チョキ」「パー」をこの 3 つのアイコンで表示する。

反復構造は、繰り返し構造ともいう。

【例題 2-3】 カウンターの利用（プログラム prei2-3）

「変数 カウンター」を利用して、表示してみよう。

「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 0 を 4 にする。

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。

注）変数は数値や文字を入れるための領域

3．プログラムの基本（分岐構造）

【例題 3-1】「グー」「パー」の表示（プログラム prei2-41）

変数 c=0 のときは、「グー」を表示し、c=1 ときは、「パー」を表示してみよう。

「最初だけ」ブロック

＊「変数」-「変数を追加する」 変数 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」を「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。

【例題 3-2】 乱数の利用（プログラム prei2-4）

乱数を利用して、変数 c に 0 から 1 の数値を入れるように変更してみよう。

「ずっと」ブロック

＊「変数」-「変数を追加する」

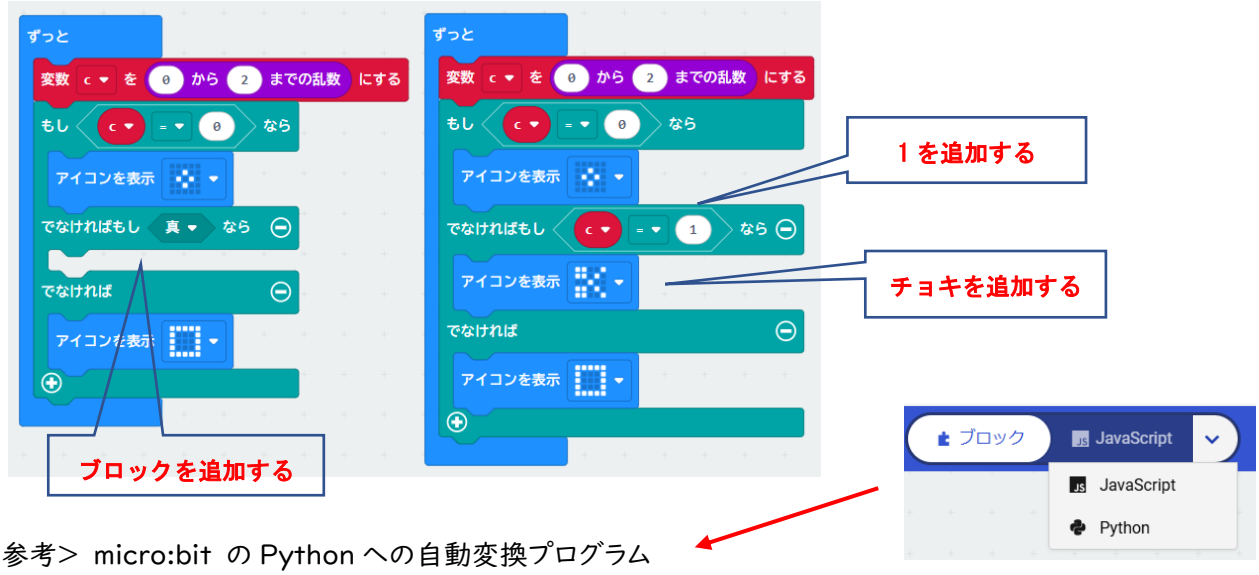
＊変数の名前を c にする。

＊「計算」-「0～10 までの乱数」 数値の 10 を 1 にする。

（乱数は 0、1 のいずれか）

【演習】「グー」「チョキ」「パー」の表示（プログラム prei2-5）

例題 3-2 のプログラムを「グー」「チョキ」「パー」を表示するように変更してみよう。



<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

【例題 4-1】「グー」「チョキ」「パー」の表示（プログラム p-rei4-1）

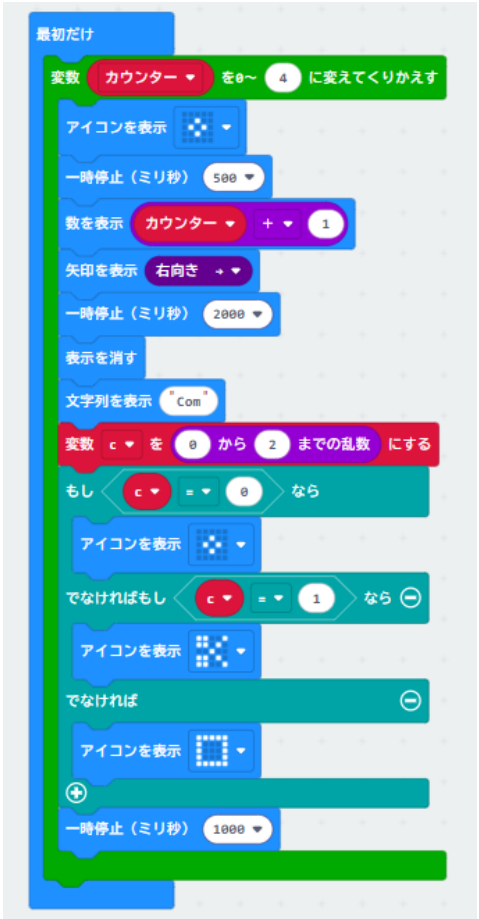
0～2 の乱数を利用して、アイコンで「グー」(数値 0)、「チョキ」(数値 1)、「パー」(数値 2)を表示してみよう(p.4 演習に同じ)。

「ずっと」ブロック

＊「変数」－「変数を追加する」 変数 c にする。

＊「計算」－「0～10 までの乱数」 数値 10 を 2 にする。

（乱数は 0、1、2 のいずれか）



【例題 4-2】コンピュータの手（プログラム prei4-2）

コンピュータの手を 5 回繰り返して、表示してみよう。

「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。

＊自分の番のときは、「矢印を表示」で「右向き」（→）」を表示して、2 秒待つ。

＊コンピュータのときは、「文字列を表示」で「Com」と表示する

＊例題 4-1 のプログラムを加え、「グー」か「チョキ」か「パー」をランダムに表示する。

【例題 4-3】(プログラム prei4-3)

スイッチボタンを利用して、自分の手をアイコンで表示してみよう。ただし、A ボタンは「グー」、B ボタンは「チョキ」、A+B ボタンは「パー」とする。

「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。

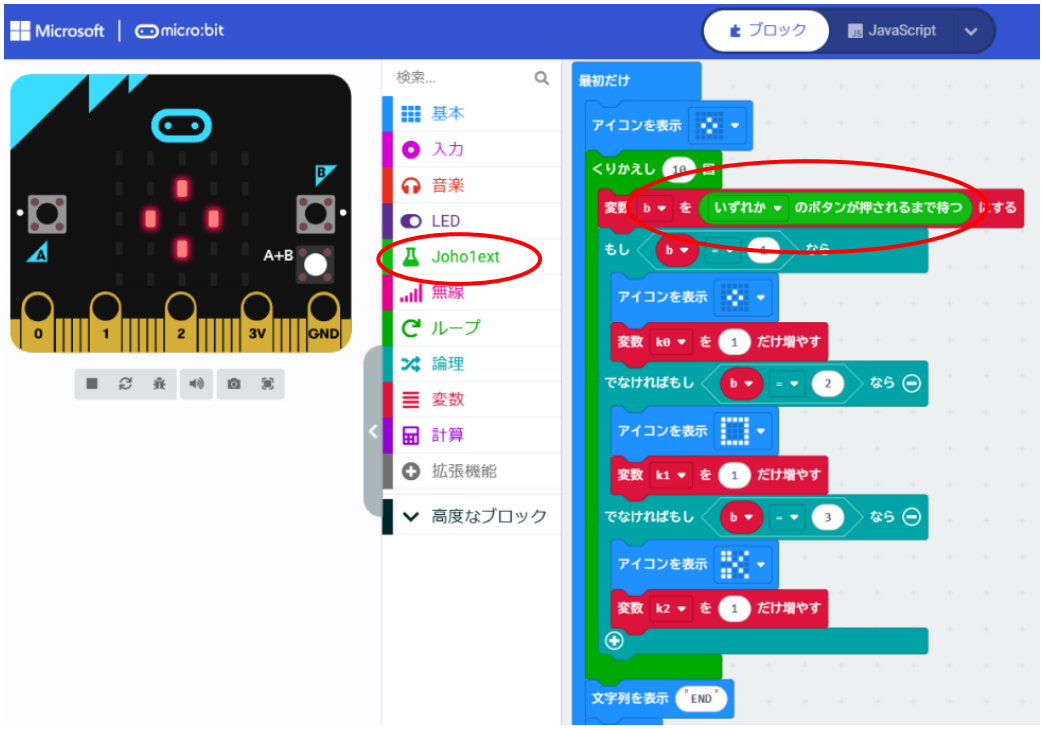
＊「入力」－「ボタン A が押されたとき」を選択する。

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

【例題 4-4】「グー」「チョキ」「パー」の出す回数(自分)(プログラム prei4-5)

自分の手を 10 回実施したとき、「グー」「チョキ」「パー」の出す回数を調べるプログラムを作成してみよう。

入力用拡張ブロック(「いずれかの ボタンが押されるまで待つ」)を利用する。



「最初だけ」ブロック

＊最初、「グー」を表示する。

＊繰り返し 10 回のループ内は、演習のプログラムと同じように、「グー」の回数の変数 k0、「チョキ」の回数の変数 k1、「パー」の回数の変数 k2 としている。

＊回数の表示は、プログラムでは、省略している。

入力用拡張ブロック（いずれかのボタンが押されるまで待つ）の戻り値（返される値）は、1、2、3に設定されており、このプログラムでの「グー」「チョキ」「パー」との対応は、下記の通りである。

A ボタンが押されたときの戻り値：	1	0（グー）
B ボタンが押されたときの戻り値：	2	1（パー）
A+B ボタンが押されたときの戻り値：	3	2（チョキ）

＜参考＞ 入力用拡張ブロック

入力用拡張ブロック（「Joho1ext」）には、つぎの 2 つのブロックがある。

・「〇〇と聞いて待つ“ ”」

数値を 0 から 9 まで変更できるブロックである。

A ボタンを押すごとに、数値が 0 から 1 ずつ増え、B ボタンを押すと確定する。

・「A ボタンが押されるまでまつ」（A、B、A+B ボタン）

ボタンが選択されるまで待つブロックである。

A ボタンを押されると「1」、B ボタンを押されると「2」、A+B ボタンを押される「3」が返される。

5．じゃんけんの自動判定

【例題 5-1】コンピュータとの対戦（プログラム prei4-6）☆

じゃんけんの自動判定（コンピュータとの対戦）するプログラムを作成してみよう。

例題 4-2 を参考にしてコンピュータの出す手の関数(Com)、例題 4-4 を参考にして自分の出す手の関数(You)を作成する。また、関数 判定を作成し、勝ったときは「うれしい顔」、負けたときは「悲しい顔」、引き分け（あいこ）のときは「困った顔」のアイコンで表示する。

関数は、以下のように作成する。

＊ツールボックスの「高度なブロック」－「関数」を選択する。

＊「関数」－「関数を作成する…」を選択すると、「関数の編集」ダイアログが表示される。

＊「関数の編集」で、まず、引数のない関数 2 つの関数「Com」と「You」を作成する。

「関数 Com」は、例題 4-2 のプログラム、「関数 You」は、「入力用拡張ブロック」を利用しており、例題 4-4 のプログラムのほぼ同じ内容である。なお、この 2 つのプログラムでは、「もし ～なら～ でなければもし～ でなければ～」の箇所は、同じ内容であるので、この箇所も関数にすることはできる。

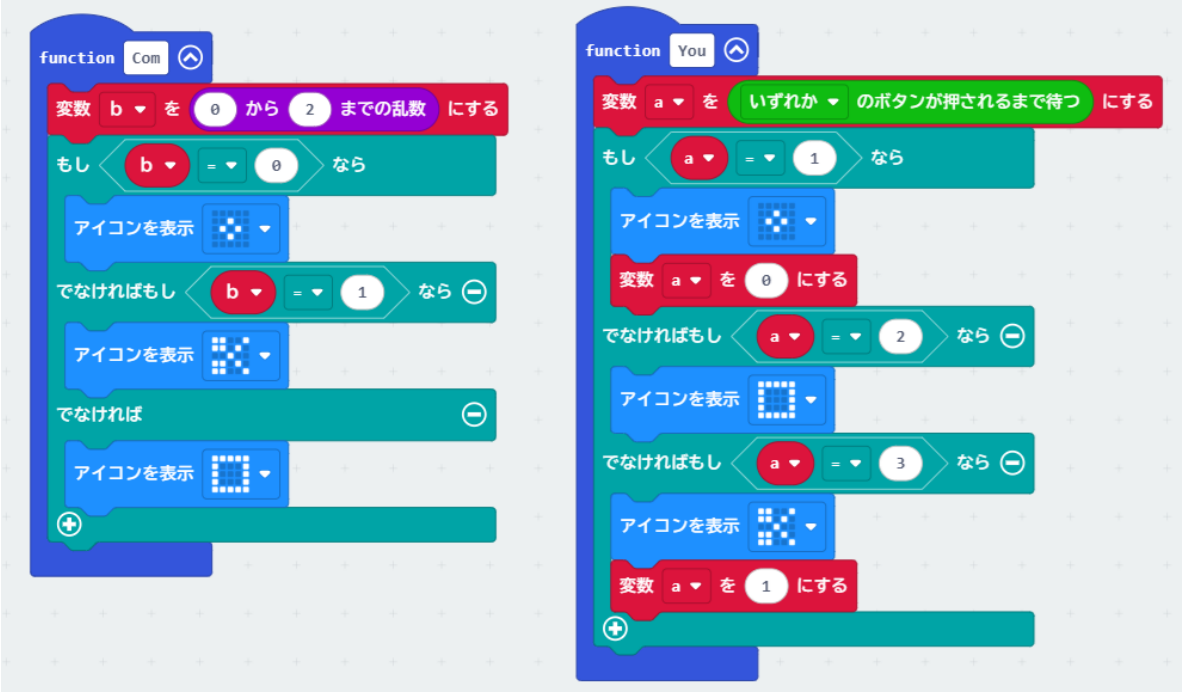
ここでは、「関数の編集」で、まず、引数のない関数 2 つの関数「Com」と「You」を作成する。

「関数 Com」

例題 4-2 のプログラムの後半部分と同じである。

「関数 You」

例題 4-4 のプログラムの後半部分と同じである。このプログラムでは、「入力用拡張ブロック（いずれかのボタンが押されたまで待つ）」を利用している。

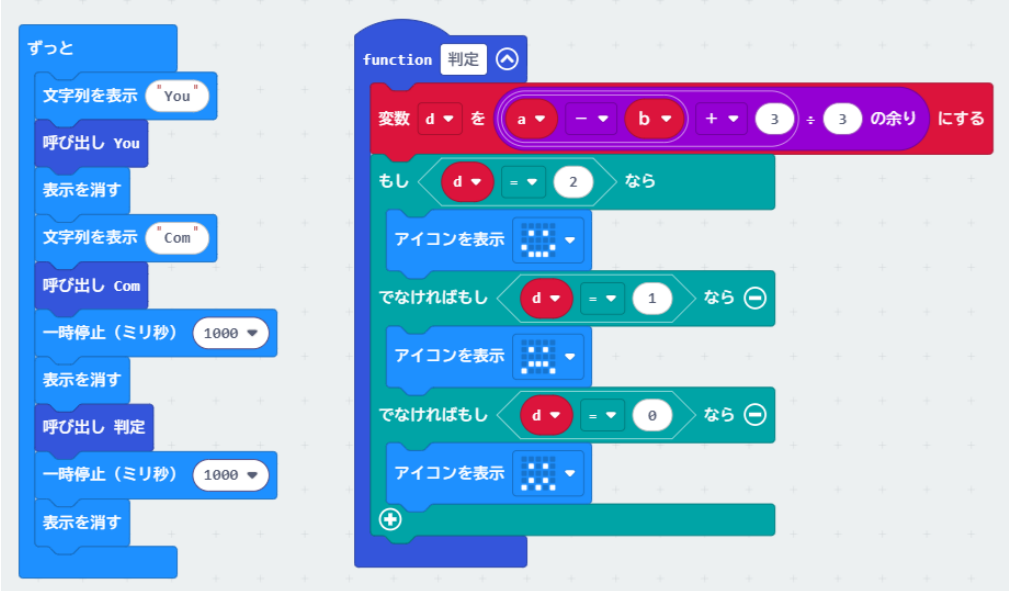


「ずっと」ブロック

＊「You」と表示し、「呼び出し You」で、関数を呼び出す。

＊「Com」と表示し、「呼び出し Com」で、関数を呼び出す。

＊「呼び出し 判定」で、関数を呼び出す。



「関数 判定」

判定式の余りを求める式の箇所は、以下のように作成する。

＊「計算」－「0 / 1 の余り」を選択し、後ろの「1」を 3 にする。

＊「計算」の「0 - 0」、「変数」の a と b から、「a - b」を作成する。

＊さらに、「計算」の「0 + 0」から「0 + 3」を作成し、「0」に「a - b」を重ねる

＊作成した「a - b + 3」を「0 / 3 の余り」の「0」に重ねる。

<勝敗の判定表>

A、B の 2 人のじゃんけんの勝敗は、下表に示す通りで、「a-b+3」を 3 で割った余りでの値で判断する。

例題 4－1 の A（You）、B（Com）の場合

引き分け：0　　A(You)の勝ち：2　　B(Com)の勝ち：1

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)

【例題 5-2】人との対戦（プログラム prei4-7）★

A さん（自分）と B さん（相手）が対戦したとき、無線通信を使って、じゃんけんを自動判定するプログラムを作成してみよう。ここでは、A ボタンを繰り返し押すと、「グー」「チョキ」「パー」を表示し、B ボタンを押すと、自分の出した「グー」「チョキ」「パー」の数値を無線で送り、自動判定をするプログラムを作成する。

「最初だけ」ブロック

＊「無線」－「無線グループの設定」を選択し、初期設定の「1」にしておく。

＊「関数 you」を呼び出して、「無線」－「Group」の「無線で数値を送信 a 」を送信する。

＊変数 d（判定の数値）、変数 a は、「0」に初期設定しておく。

＊自分と相手の画面が表示され、a が「0」なので、「グー」が表示される。

注）micro:bit 2 台に、同じプログラムをダウンロードする。同時にいくつかのグループで行うときは、無線のグループ番号を変え、2 台以外は同じ番号にしないこと。

「無線で受信したとき」

＊「無線」－「Receive」の「無線で受信したとき receivedNumber」を選択する。

＊変数 b を「receivedNumber」にする。この変数名「receivedNumber」を、ドラッグ＆ドロップで入力する。

「ボタン A」

＊余りを求める計算式を利用して、変数 a に、0（A ボタンで「グー」）、1（A+B ボタンで「チョキ」）、2(B ボタンで「パー」)を代入する。

＊「関数 you」を呼び出して、選択した数値（変数 a）を送信する。

「ボタン B」

＊「関数 判定」を呼び出して、勝敗をアイコンで表示する。

このプログラムでは、A ボタンを押すごとに、「グー」「チョキ」「パー」の順で変わり、変数 a の数値は、0、1、2 と増えるので、「関数 You」の変数 a の値も同じである。「関数 判定」は、例題 5－1 のプログラムと同じである。

注）シミュレーション画面で実行が確認できれば、2 台の micro:bit にプログラムをダウンロードして確かめてみる。