micro:bit で学ぶプログラミング

～ブロック型，JavaScript，MicroPython～

日本情報科教育学会

第 12 回全国大会（ワークショップ）

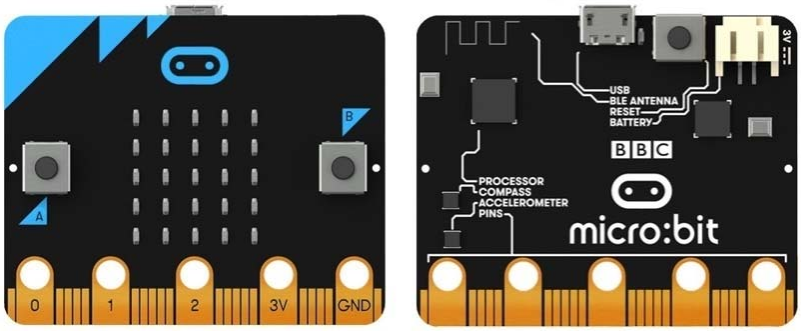
日時： 2019 年 7 月 20 日(日)　10:45～11:45
場所： 北九州市立大学

担当： 高橋　参吉（NPO 法人　学習開発研究所）
稲川　孝司（帝塚山学院大学非常勤）
喜家村　奨（帝塚山学院大学）

目　次

１．micro:bit の基本操作

micro:bit は，下記の図のような手のひらサイズのコンピュータです[1),2)]。



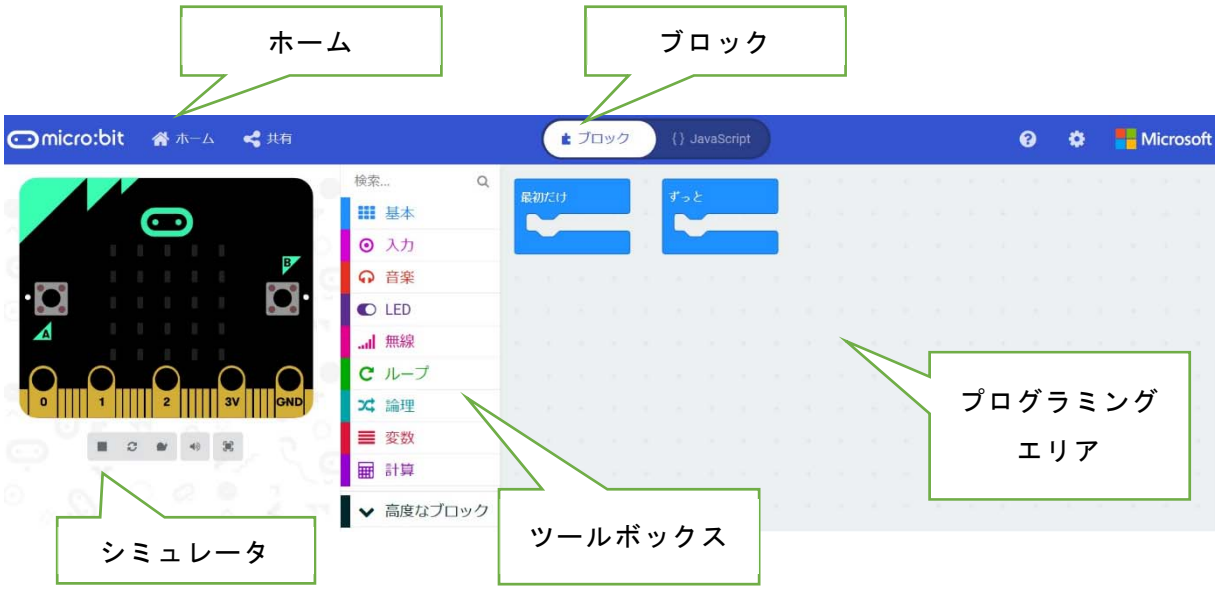
下記の micro:bit の Web サイトへアクセスします。

https://makecode.microbit.org/

その中に，「マイプロジェクト」，「チュートリアル」，「ゲーム」などがあり，「マイプロジェクト」では，新しいプロジェクトを作成したり，保存したプロジェクト（プログラム）を読み込むことができます。

新しいプロジェクトを選択すると，次の図のような micro:bit のシミュレータ画面が表示されます。

＊日本語で表示されていない場合は，Web ページの一番下で日本語を選択しておく。



画面の左側から，micro:bit での実行が確認できるシミュレータ，ツールボックス，そして一番右側が，プログラミングエリアです。

［ツールボックス］

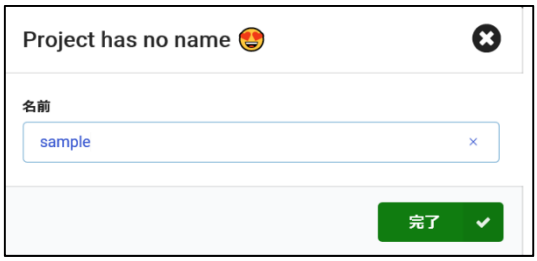
ツールボックスには，基本，入力，音楽，LED，無線，ループ，論理，変数，計算，そして，高度なブロックがあり，それぞれのツールボックスのツールをクリックすると，利用できるブロックが表示されます。

［プログラミングエリア］

プログラミングエリアは，ツールボックスで選択したブロックをエリア内にドロップすることによってプログラムが書けるブロックエディタになっています。最初に，「最初だけ」，「ずーと」のブロックが置かれています。

［ホーム］

画面の上部左側には「ホーム」があり，「ホーム」を選択すると，新しいプロジェクトに名前を付けて保存できます。



［ブロック］［JavaScript］

画面の上部の真ん中の「ブロック」を「JavaScript」に切り替えることによって，「ブロック」で描かれたプログラムを「JavaScript」で表示することができます。

［ダウンロード］

画面の下に，［ダウンロード］「題名未設定…」「アイコン（FD)」というボックスがあり，プログラム名を入れてパソコンにプログラムを保存したり，micro:bit にプログラムをダウンロードしたりすることができます。名前を入れて，右にある「アイコン（FD)」をクリックすると，パソコンのフォルダにファイルを保存することができます。

また，パソコンの USB に micro:bit をつないで，次の表示に従って micro:bit にプログラムを転送することができます。

転送が完了すれば，micro:bit でプログラムを動かすことができます。

もし，micro:bit にプログラムが格納されていれば，上書きされるので注意しょう。

［シミュレータ］

また，micro:bit のシミュレータは，micro:bit の画面の下のボタンが，四角ボタン（■）であれば，クリックすると開始，三角ボタン（▶）であれば，クリックすると停止できます。

それでは，次の例題で，micro:bit の基本操作を確かめてみましょう。

【例題 1－1】

図のようなハートマークを LED 画面に表示させよう。次に，ハートマークを点滅させてみよう。作成したプログラムはパソコンに保存する。

＜手順 1＞（ファイル名：rei1-1-1）

1) 「ホーム」をクリックし，「新しいプロジェクト」を選択する。
2) ツールボックスの中の「基本」をクリックし，「LED 画面に表示」ブロックをドラッグ＆ドロップでプログラミングエリアに移動する。
3) 「最初だけ」ブロックに「LED 画面に表示」ブロックをつなげる。
4) LED をクリックすると光の ON/OFF が切り替わるので，ハート形に見えるように LED を ON にし，動作を確認する。
   なお，不要なブロックは，ツールボックスへドラッグ＆ドロップすると削除できる。

＜手順 2＞（ファイル名：rei1-1-2）

1) 次に，「最初だけ」ブロックを「ずっと」ブロックに変更する。
2) 「基本」から「一時停止（ミリ秒)」ブロックをつなぎ，数値を 100 から 500 に変えておく。
3) 「基本」から「表示を消す」ブロックをつなぐ。
4) 「一時停止（ミリ秒)」ブロックをつなぎ，数値を 100 から 500 に変えておく。
5) ダウンロードの右のアイコンをクリックして，適切なフォルダにプログラム名（ファイル名を rei○○とすると，実際には microbit-rei○○.hex となる）をつけて，パソコンに保存する。
6) パソコンの USB に micro:bit をつなぎ，保存したプログラムを選んで，右クリック⇒送る⇒MICROBIT　でファイルを転送できる。
7) 転送している間，mocro:bit の裏側の LED がオレンジ色に点滅し，点滅が終われば，リセットボタンを押して，プログラムを起動させる。

２．プログラムの基礎（順次，繰返し）

【例題 1－2】　次のプログラムを作成して，図のように表示されることを確かめよう。(ファイル名：rei1-2)

＜手順＞

1)　「基本」から「最初だけ」ブロックを選択する。

2)　「LED」から「点灯」ブロックを選択し，x を 2，y を 0 にする。なお，左上の LED の座標は（0，0），右下の LED の座標は（4，4）である。

3)　「点灯」ブロックにマウスをあて，マウスの右ボタンを押して複製を選択する。

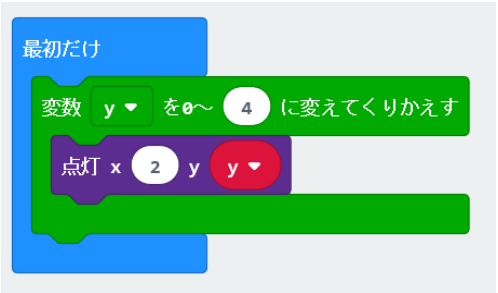
4)　点灯ブロックを 4 回コピーし，x をすべて 2，y を 1～4 にする。

5)　5 つの点灯ブロックを「最初だけ」ブロックに接続し，動作を確認する。

このようなプログラムの構造を順次構造という。

【例題 1－3】　例題 1－2 のプログラムを，「ループ」から，繰り返しのブロックを使って，プログラムを変更してみよう。(ファイル名：rei1-3)

＜手順＞



1)　点灯ブロック（5 つ）を「最初だけ」ブロックから外す。

2)　「ループ」から「変数（カウンター）を 0～4 に変えてくりかえす」ブロックを選択する。

3)　「変数（カウンター)」の箇所を選択した後，「変数の名前を変更」を選択して，ダイアログが表示されるので「y」に変更する。

4)　「最初だけ」ブロックに接続する。

5)　「LED」から「点灯」ブロックを選択し，x の「0」を「2」に変更する。

6)　「変数」から「y」を選択し，「点灯」ブロックの y の「0」の上に置く。

7)　変更した「点灯」ブロックを「変数 y を 0～4 に変えてくりかえす」ブロックに接続する。

変数は，数値や文字などのデータを入れる容器に当たるものであり，数値や文字などのデータを定数という。このようなプログラムの基本構造を繰返し構造という。

【例題 1－4】　例題 1－3 で，点灯の x 座標を変数 x，y 座標を変数 4-x に変更して，図の形を確認してみよう。次に，作成したプログラムが「JavaScript」では，どのように書かれているか確認してみよう。（ファイル名：rei1-4）

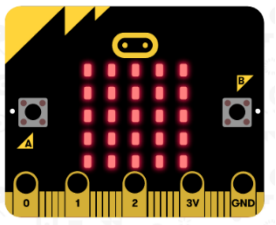
＜手順（概略）＞

1)　「計算」から，「引き算」のブロックを選択する。

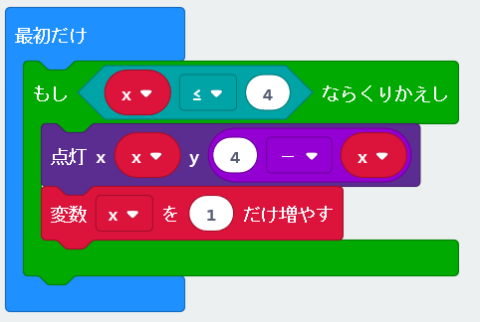
2)　計算式（4-x）を作成して，「点灯」の y 座標に置く。

JavaScript のプログラムでは，繰返しは for を使う。また，x++は， x を 1 ずつ増やすことである。

【例題 1－5】　次のプログラムでは，LED がすべて点灯する図形になるか，点灯の順序も確かめてみよう。（ファイル名：rei1-5）



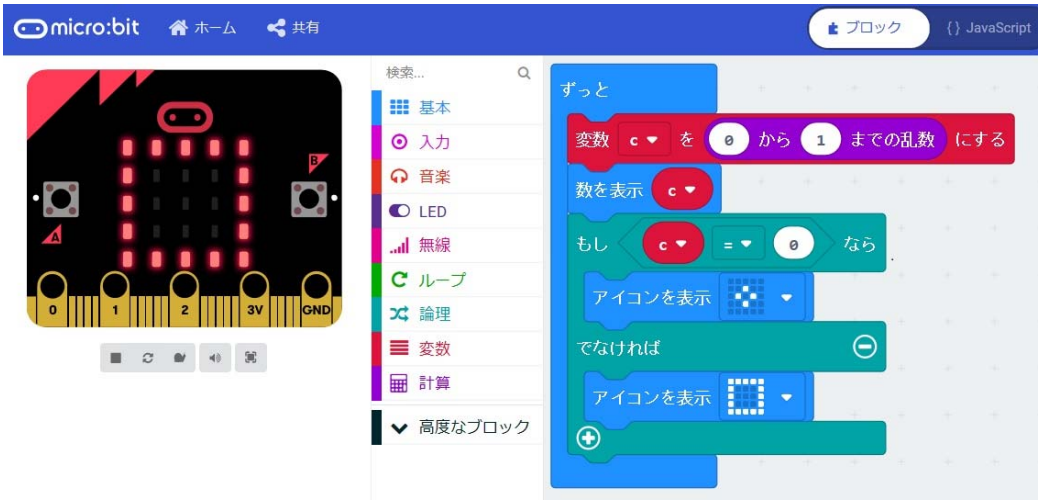
【例題 1－6】　例題 1－3 のプログラムの繰り返し「for～」が「while～」になるように，「ループ」の箇所でブロックを変更して，プログラムを作成しよう。また，JavaScript のプログラムを確認してみよう。（ファイル名：rei1-6）



「x += 1」は，「 x = x + 1 」と同じ内容で，最初の「let x = 0」，ループ内にある「x += 1」で，x を 1 ずつ増やし，繰り返していくことになる。なお，くりかえし回数がわからない時は，for は使えないので While を使うとよい。

３．プログラムの基礎（分岐）

【例題 1－7】　乱数（0，1）を発生させて，変数 c に代入して，c が 0 の時は「小さいダイアモンド」（グー），c が 1 の時は「しかく」（パー）を「ずっと」くりかえし表示するようなプログラムを作成しよう。（ファイル名：rei1-7）



＜手順＞

1)　「基本」から「ずっと」ブロックを選択する。

2)　「論理」から「もし～なら～でなければ」ブロックを選択する。

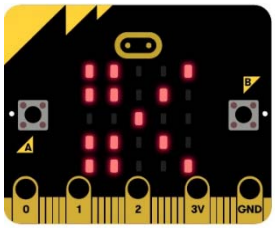
3)　「基本」から「「アイコン表示」ブロックを選択し，「小さいダイアモンド」「しかく」を選択する。「小さいダイアモンド」は「～なら」，「しかく」は「～でなければ」の後に接続しておく。

4) 「変数」から「変数」ブロックを選択し，変数の名前を c にする。また，「変数を 0 にする」ブロックを選択して，変数の箇所を c に変更する。

5) 「論理」から「 0＝0 」ブロックを選択し，「 c＝0 」に変更し，「もし・・・」ブロックに重ねる。

6) 「計算」から「0～4 の範囲の乱数」を選択し，範囲を「0～1」にし，「変数・・・」ブロックに重ねる。
このようなプログラムの構造を分岐構造という。

【例題 1－8】



乱数（0，1，2）を発生させて，変数 c に代入して，c が 2 の時は，「アイコン表示」の「はさみ」（チョキ）を表示するようなプログラムに変更しよう。（ファイル名：rei1-8）

＜手順＞

1) 「もし～なら～でなければ」ブロックを「もし～なら～でなければもし～なら～でなければ」とするには，ブロックの「＋」マークをクリックすると，「でなければもし～なら」が追加される（図(a)）。次に，グーを移動，チョキを追加する（図(b)。

2) 「0～1 の範囲の乱数」を「0～2 の範囲の乱数」にしておく。

3) 「でなければもし」の箇所に，「 c ＝ 1 」にブロックを追加しておく。

(a)変更前

(b)変更後

次ページ以降に，例題 1-1 から例題 1-8 の JavaScript，MicroPython のプログラムを示す。解説は，講習中に行う。なお，プログラムのインデント(字下げ)は，自動インデントではスペース 4 つだが，紙面の関係でスペース 2 つにしている。

４． JavaScript，MicroPython のプログラム

＜JavaScript＞

1） micro:bit の基本操作

【例題 1-1】
(保存ファイル名：microbit-rei1-1-1)
```
basic.showLeds(`
    . # . # .
    # # # # #
    # # # # #
    . # # # .
    . . # . .
    `)
```

(保存ファイル名：microbit-rei1-2)
```
basic.forever(function () {
  basic.showLeds(`
    . # . # .
    # # # # #
    # # # # #
    . # # # .
    . . # . .
  `)
  basic.pause(500)
  basic.clearScreen()
  basic.pause(500)
})
```

2） プログラムの基礎（順次，繰返し）

【例題 1-2】
(保存ファイル名：microbit-rei1-2)
```
led.plot(2, 0)
led.plot(2, 1)
led.plot(2, 2)
led.plot(2, 3)
led.plot(2, 4)
```

【例題 1-3】
(保存ファイル名：microbit-rei1-3)
```
for (let y = 0; y <= 4; y++) {
  led.plot(2, y)
}
```

【例題 1-4】
(保存ファイル名：microbit-rei1-4)
```
for (let x = 0; x <= 4; x++) {
  led.plot(x, 4 - x)
}
```

＜MicroPython＞

1） micro:bit の基本操作

【例題 1-1】
(保存ファイル名：microbit-py-rei1-1-1)
```
from microbit import *

Heart = Image("09090:99999:99999:09990:00900")
display.show(Heart)
```

(保存ファイル名：microbit-py-rei1-1-2)
```
from microbit import *

Heart = Image("09090:99999:99999:09990:00900")

while True:
  display.show(Heart)
  sleep(500)
  display.clear()
  sleep(500)
```

2） プログラムの基礎（順次，繰返し）

【例題 1-2】
(保存ファイル名：microbit-py-rei1-2)
```
from microbit import *

display.set_pixel(2, 0, 9)
display.set_pixel(2, 1, 9)
display.set_pixel(2, 2, 9)
display.set_pixel(2, 3, 9)
display.set_pixel(2, 4, 9)
```

【例題 1-3】
(保存ファイル名：microbit-py-rei1-3)
```
from microbit import *

for y in range(0, 5):
  display.set_pixel(2, y, 9)
```

【例題 1-4】
(保存ファイル名：microbit-py-rei1-4)
```
from microbit import *

for x in range(0, 5):
  display.set_pixel(x, 4-x, 9)
```

【例題 1-5】
(保存ファイル名：microbit-rei1-5)

```
for (let x = 0; x <= 4; x++) {
    for (let y = 0; y <= 4; y++) {
        led.plot(x, y)
        basic.pause(100)
    }
}
```

【例題 1-6】
(保存ファイル名：microbit-rei1-6)

```
let x = 0
while (x <= 4) {
    led.plot(x, 4 - x)
    x += 1
}
```

3） プログラムの基礎（分岐）

【例題 1-7】
(保存ファイル名：microbit-rei1-7)

```
let c = 0
basic.forever(function () {
    c = Math.randomRange(0, 1)
    basic.showNumber(c)
    if (c == 0) {
        basic.showIcon(IconNames.SmallDiamond)
    } else {
        basic.showIcon(IconNames.Square)
    }
})
```

【例題 1-8】
(保存ファイル名：microbit-rei1-8)

```
let c = 0
basic.forever(function () {
    c = Math.randomRange(0, 2)
    basic.showNumber(c)
    if (c == 0) {
        basic.showIcon(IconNames.SmallSquare)
    } else if (c == 1) {
        basic.showIcon(IconNames.Scissors)
    } else {
        basic.showIcon(IconNames.Square)
    }
})
```

【例題 1-5】
(保存ファイル名：microbit-py-rei1-5)

```
from microbit import *

for x in range(0, 5):
    for y in range(0, 5):
        display.set_pixel(x, y, 9)
        sleep(100)
```

【例題 1-6】
(保存ファイル名：microbit-py-rei1-6)

```
from microbit import *

x = 0
while x <= 4:
    display.set_pixel(x, 4-x, 9)
    x += 1
```

3） プログラムの基礎（分岐）

【例題 1-7】
(保存ファイル名：microbit-py-rei1-7)

```
from microbit import *
import random

while True:
    c = random.randint(0, 1)
    display.scroll(str(c))
        if c == 0:
            display.show(Image.DIAMOND_SMALL)
        else:
            display.show(Image.SQUARE)
        sleep(500)
```

【例題 1-8】
(保存ファイル名：microbit-py-rei1-8)

```
from microbit import *
import random

Scissors = Image("99009:99090:00900:99090:99009")

while True:
    c = random.randint(0, 2)
    display.scroll(str(c))
        if c == 0:
            display.show(Image.DIAMOND_SMALL)
        elif c == 1:
            display.show(Scissors)
        else:
            display.show(Image.SQUARE)
        sleep(500)
```

５．プログラムに対する注意事項

本テキストで利用している例題プログラムなどは，NPO 法人学習開発研究所の下記の Web サイトからダウンロードしてください。

本書の中で記載している JavaScript のファイル名，例えば，rei〇〇は，保存ファイル名では，microbit-rei〇〇.hex，　になっています。

http://www.u-manabi.org/microbit/

一方，MicroPython のファイル名については，rei〇〇を hex ファイルにしたときに，MicroPython のファイルであることを識別するために，少し長いですが，microbit-py-rei〇〇としています。したがって，hex ファイルは，microbit-py-rei〇〇.hex になります。

micor:bit では，ブロックから JavaScript へ自動変換されますが，JavaScript の変数や関数の名称・順序は，自動変換されたプログラムと異なる場合があります。

MicroPython への変換については，できるだけ，JavaScript のプログラムと互換性を持たせるようにしていますが，文法が異なることもあり，プログラムが異なっている個所があります。micro:bit や Python の特徴を生かしたプログラムの記述[3)-5)]もあると思いますが，皆さんで，検討していただければ幸いです。

エディタについては，micro:bit にはブラウザ上でプログラムの開発が可能な Python の開発環境[6)]も用意されていますが，参考 Web サイトで紹介されている「mu エディタ」[7)]をダウンロードして，「BBC micro:bit」(MicroPython) 編集用を利用すると便利です。

参考文献，参考 Web サイト

1) micro:bit の公式 Web サイト（日本語）：https://microbit.org/ja/
   micro:bit の冒険を始めよう　https://microbit.org/ja/guide/
2) ガレス・ハルファクリー著，金井哲夫訳：BBC マイクロビット公式ガイドブック，日経 BP 社(2018.10).
3) BBC micro:bit MicroPython ドキュメンテーション
   https://microbit-micropython.readthedocs.io/ja/latest/index.html
4) チュートリアル
   https://microbit-micropython.readthedocs.io/ja/latest/tutorials/introduction.html
5) Simon Monk:Programming the BBC micro:bit: Getting Started with MicroPython, McGraw-Hill Education(2017.11).
6) Micro:bit-Python editor
   https://python.microbit.org/
7) mu エディタ (Code with Mu: a simple Python editor for beginner programmers.)
   https://codewith.mu/