

# プログラミング

あ ・ そ ・ ぼ



いまをつくる！

未来を創る！

NPO法人 NEXTDAY

## n:bit



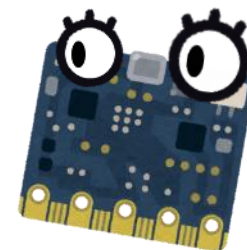
& micro:bit

## micro:bit入門

(基礎講座)



n:bit  
ファミリー



2025年12月7日(日)

まなビ〜クラブ

主催：きたひろしま西の里地区生涯学習振興会  
協力：NPO法人NEXTDAY

NPO法人 NEXTDAY



<https://nextday-kids.com/>

Ver.2025.01

# はじめてのmicro:bit v2

45

全員で同じ内容  
を組み組みます。

- Step-1 準備しよう
- Step-2 使ってみよう
- Step-3 センサーの値を表示しよう
- Step-4 ボタン A・B / ロゴをタップ(タッチ)

n:bit  
& micro:bit

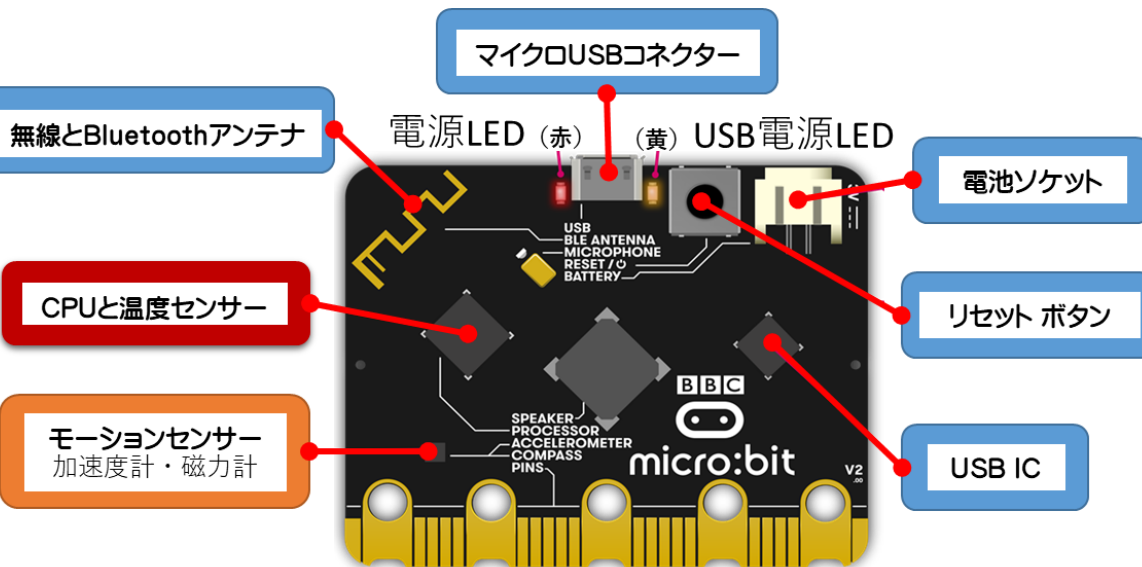
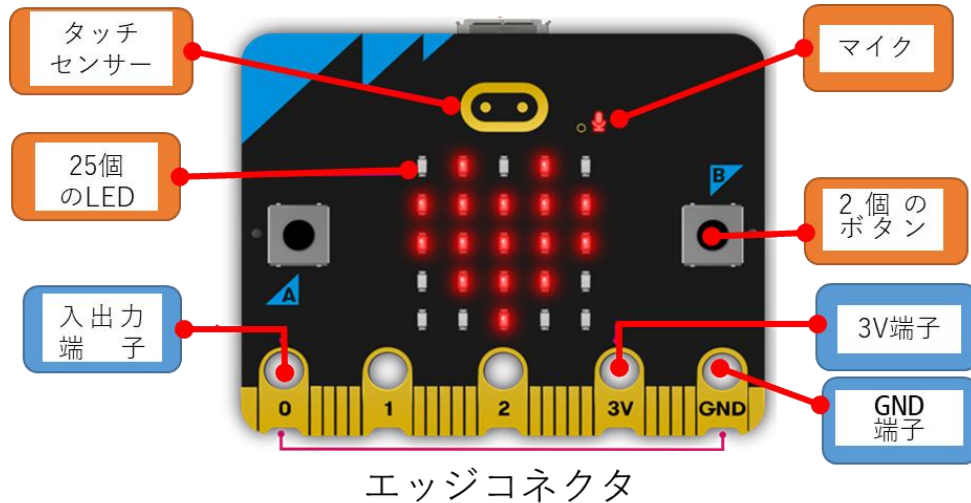
50

選択課題から  
自由を選んで  
取り組みます。



- Step-5 <sup>カラー</sup>Color LEDを光らせよう 単色/虹色/方角
- Step-6 コンパス(方位磁針)をつくろう ~磁気センサー~
- Step-7 サイコロをつくろう その1/その2
- Step-8 じゃんけんゲームをつくろう ~加速度センサー~
- Step-8 磁力計でお宝をさがせ グラフ/音 ~磁気センサー~
- Step-9 歩行者信号付き信号機をつくろう

# micro:bit



## 特徴

- ・ブロック型とスクリプト型プログラミング
- ・シミュレーター
- ・多様なセンサと機能が内蔵
- ・多彩な活用と拡張（創造性）
- ・通信機能

## n:bit & micro:bit



# はじめてのmicro:bit

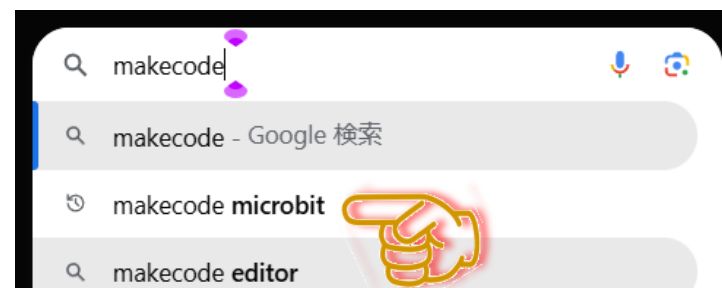
## ・オンライン版makecode（無料）

インターネットに接続しているパソコンのブラウザ  
（Microsoft Edge または Google Chrome を推奨）でが使えます。

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

## ・使って(動かして)みよう

●Google検索 **makecode** と入力  
**makecode microbit** を選択



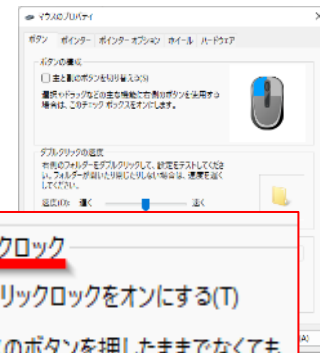
### 誤操作・誤動作防止対策

#### 1. 意図しないwindows操作防止

一> Makecode画面で  
[F11：全画面]モードにする

#### 2. クリックロック防止

一> マウスのプロパティ  
でクリックロック機能解除



### チュートリアル

プログラムの作り方を自分で学べる



The screenshot shows the Microsoft MakeCode editor interface for micro:bit. The interface is divided into several sections:

- Toolbar (Top):** Includes the micro:bit logo, "プロジェクト" (Project), "共有" (Share), "ブロック" (Blocks), and "JavaScript" tabs. There are also icons for help, settings, and the Microsoft logo.
- Tool Box (Left):** A vertical menu with categories: "基本" (Basic), "入力" (Input), "音楽" (Music), "LED", "変数" (Variables), "計算" (Math), and "高度なブロック" (Advanced Blocks).
- Simulation Area (Left):** A visual representation of the micro:bit board with pins labeled 0, 1, 2, 3V, and GND.
- Block Programming Area (Right):** A workspace for building programs using blocks. A blue block labeled "ずっと" (Forever) contains a "文字列を表示" (Show String) block with the text "Hello!".
- JavaScript Editor (Right):** A text editor for writing JavaScript code. The code shown is:

```
1 basic.forever(() => {
2   basic.showString("Hello!")
3 })
4
```
- Bottom Bar:** Contains a "ダウンロード" (Download) button, a "名前未設定" (Name not set) field, and a "保存" (Save) button.

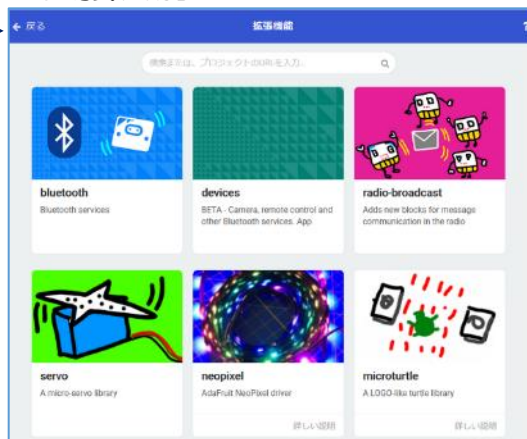
Annotations and labels are overlaid on the image:

- プログラミングエリア** (Programming Area) points to the block programming area.
- ブロックプログラム** (Block Program) points to the "Hello!" block.
- JavaScriptプログラム** (JavaScript Program) points to the JavaScript code editor.
- ツールボックス** (Tool Box) points to the left-hand menu.
- シミュレーター** (Simulator) points to the micro:bit board visualization.
- ダウンロード** (Download) points to the "ダウンロード" button.
- micro:bitに転送** (Transfer to micro:bit) points to the "ダウンロード" button.
- 名前未設定** (Name not set) points to the name input field.
- プログラム名** (Program Name) points to the name input field.
- 保存** (Save) points to the "保存" button.
- パソコンに保存** (Save to PC) points to the "保存" button.

プログラムで制御/操作する命令ブロックが種類（色）別に整理されています。



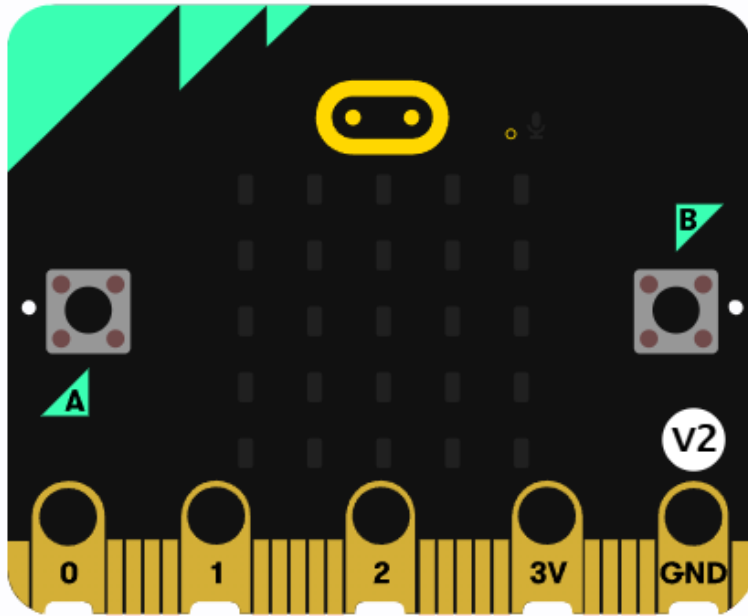
## ・拡張機能



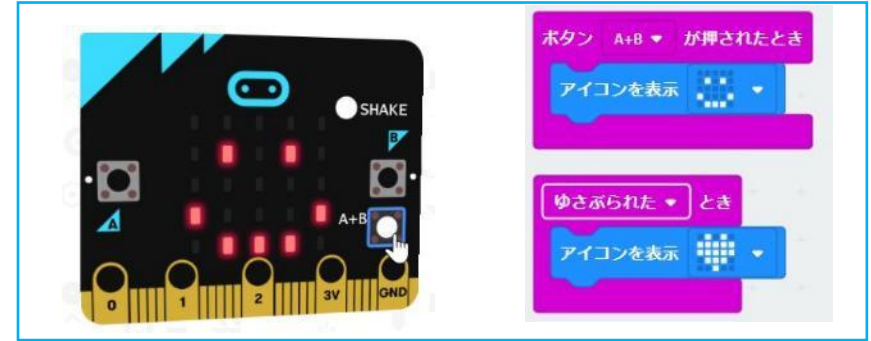
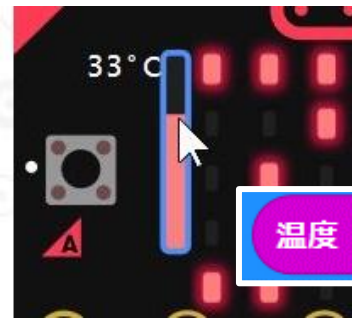
周辺機器専用のブロックが各メーカーから提供されています。それらを使う場合、この拡張機能からブロックを追加できます。

より複雑な制御や機能を実現するための追加ブロックです。複数の機能を組み合わせて

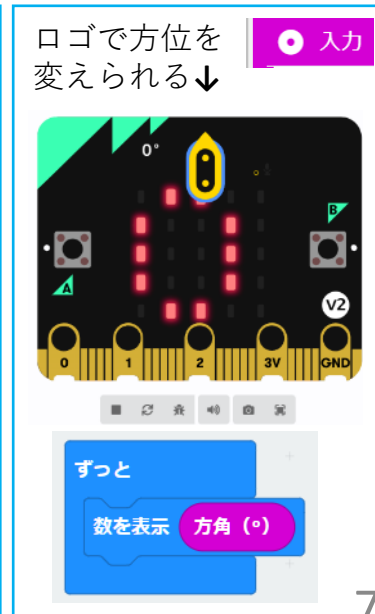
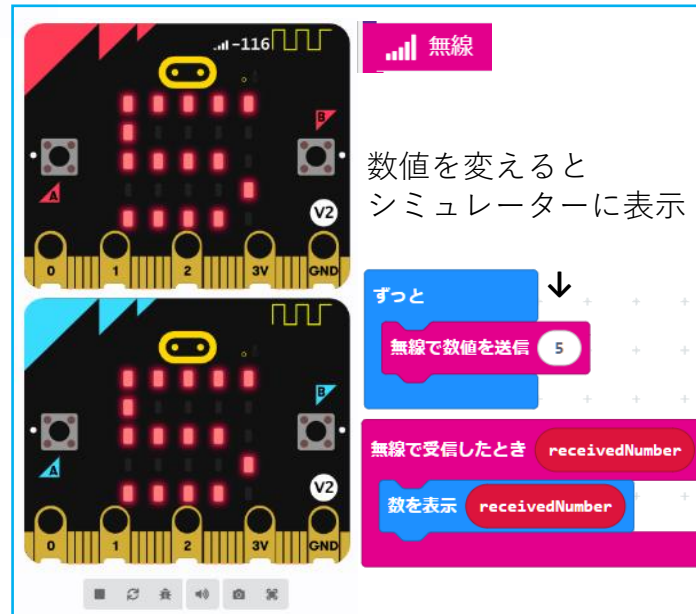




micro:bit本体がなくてもプログラムを動作させて確認することができます。使用した命令（ブロック）によって自動的に変化します。マウスで操作したり、センサーの数値を変化させることができます。



↑マウスでボタンを押したり  
ゆさぶることができる



# S1\_1 Microsoft MakeCode for micro:bit で新しいプロジェクトをつくる

① 新しいプロジェクトをクリック

② プログラムの名前を入力

③ 作成をクリック

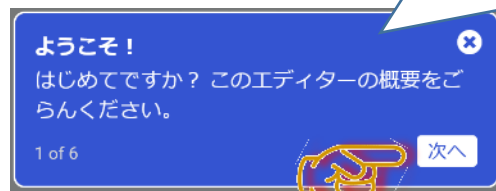
プロジェクトを作成する 🎉

プロジェクトに名前をつけてください。

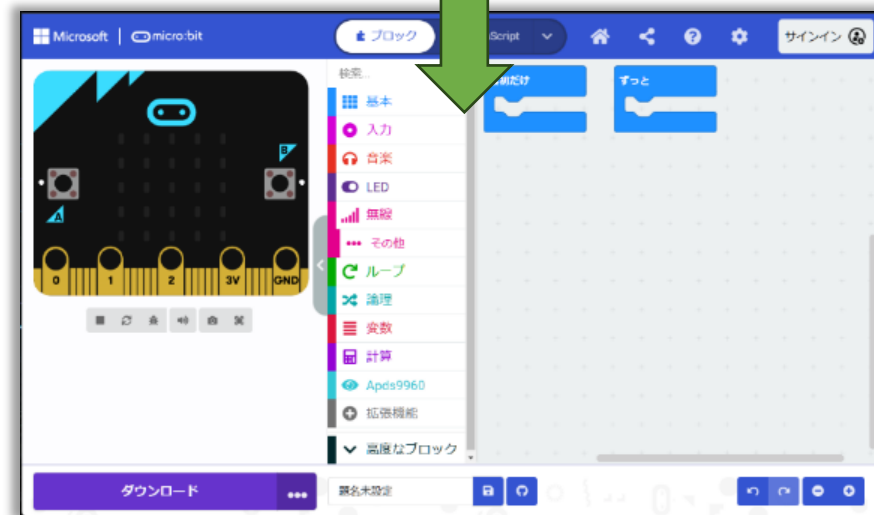
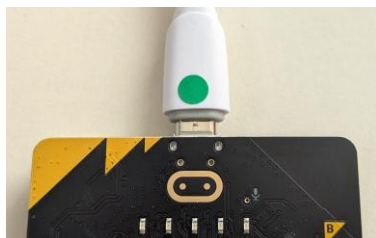
sasaki

作成 ✓

もし、“ようこそ!”のポップアップが出たら **次へ**をクリックして確認しよう。



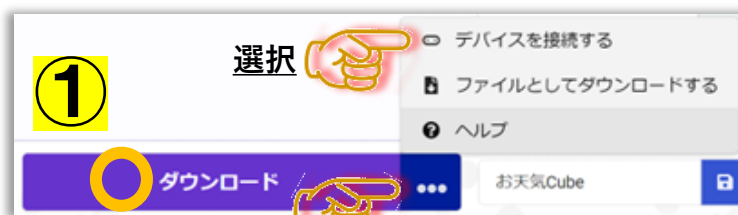
マイクロUSB端子には、表裏があります。よく見てmicro:bitにさしましょう。



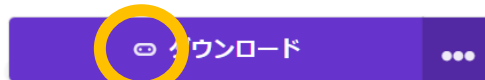
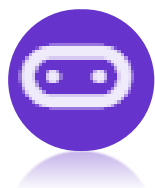
※micro:bitのプログラム(プロジェクト)は、MakeCodeエディターを使って作成します。ここでは必要最小限の作業を説明しています。詳細については、BBC公式ホームページのクイックスタートを参照してください。 <https://archive.microbit.org/ja/guide/quick/>



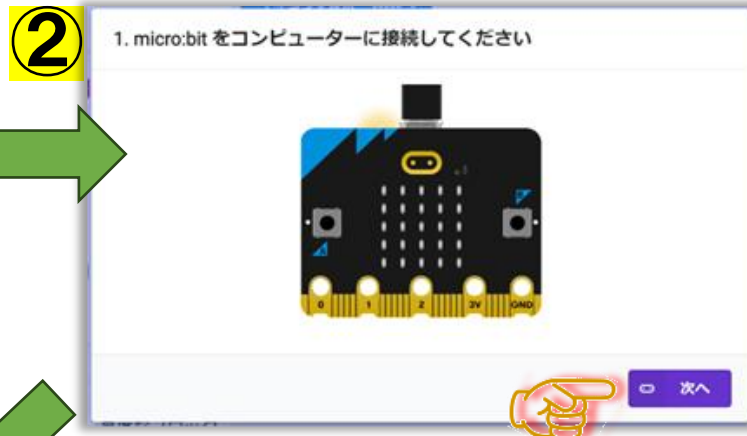
## S1\_2 micro:bitを接続する



↑ ロゴが表示されていない



ロゴが表示されたら、  
接続済みです。  
このペアリングは不要です。



接続中を示すロゴを確認



# s1\_3 プログラムをダウンロード（micro:bitに書き込み）する

## ① ロゴを確認



ロゴが非表示の場合は、  
パソコン側またはmicro:bit側の  
ケーブルをつなぎなおしてみましょう。  
S1-2の手順でペアリングをしないおすと  
接続できる場合があります。

## ② ロゴの 点めつが終わってから、 ダウンロードをクリック



## ③ プログラムのダウンロード中 ダウンロードが終わるまで ケーブルはぬかない



ダウンロード中は  
LEDが点滅します



## ④ ダウンロードが終わったことを確認



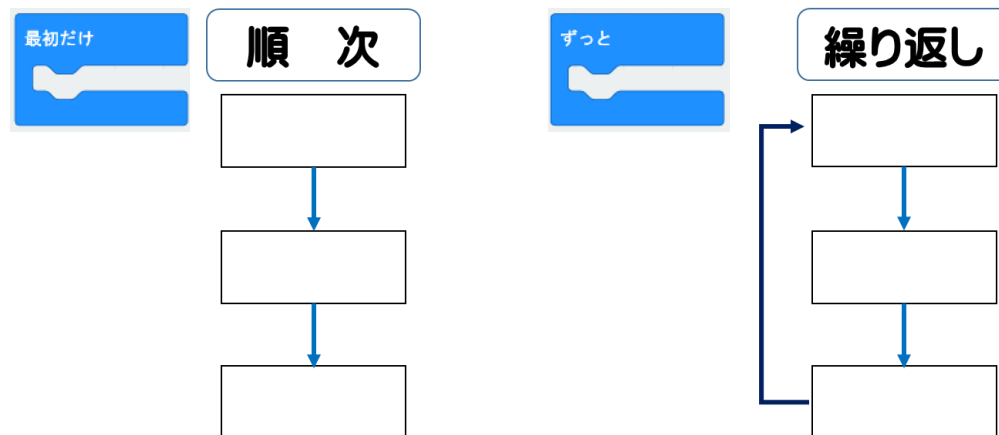
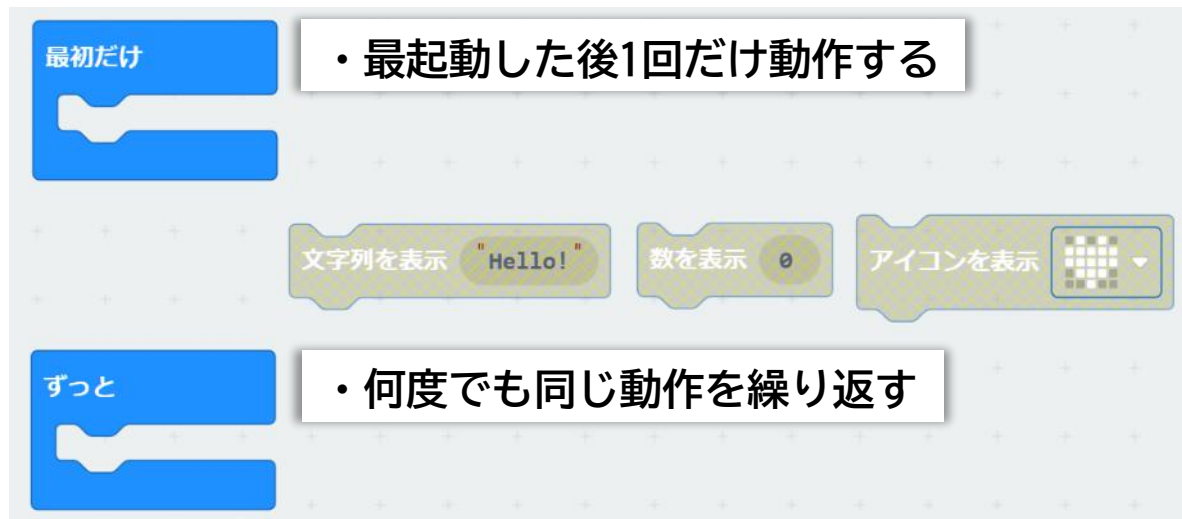
※なにもプログラムをつくずダウンロードしたので、  
micro:bitは動作しません。

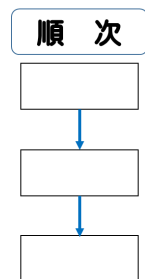
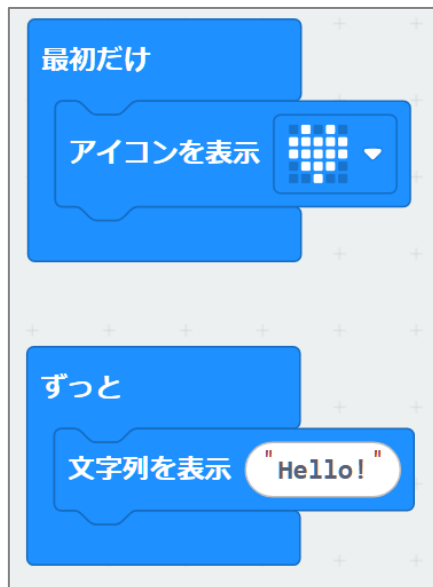
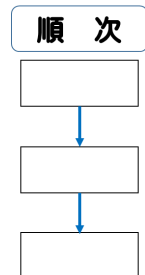
## Step 2

# 文字 / 数字 / アイコン を 表示しよう

- ・「Hello!」を表示しよう
- ・自分の名前をローマ字で表示しよう

ツールボックス > 基本



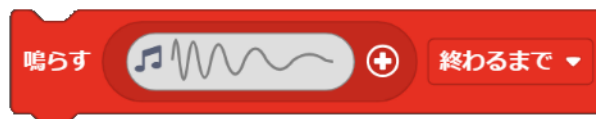




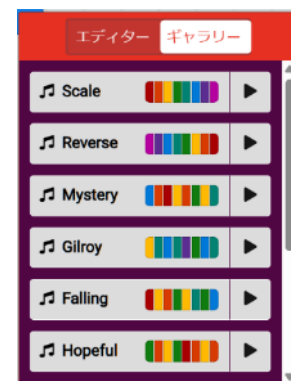
# 音をならそう

音楽ブロックをつかい、音や音楽、ゲームの効果音など表現できます。

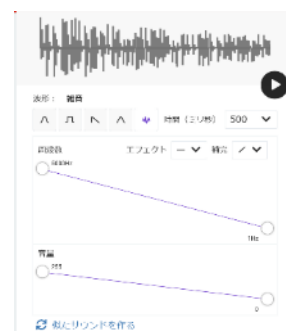
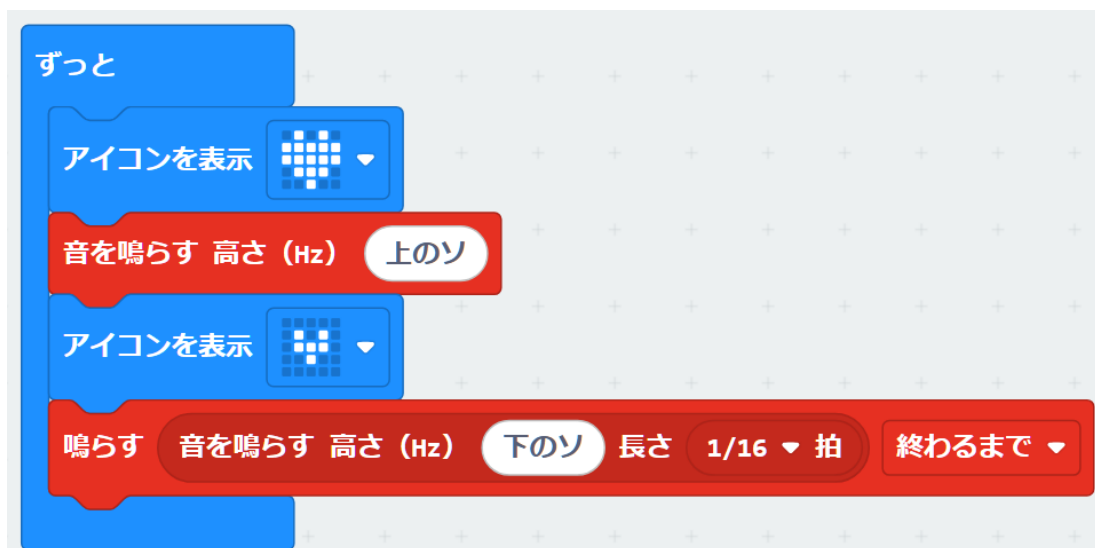
## 音楽



自分で音や効果音をつくったり、用意されてるの音をならすことができます。



プログラム例) 心臓の鼓動のように音をならす



## Step 3

# センサーの値を表示しよう

ツールボックス

> 入力

明るさ

方角 (°)

温度 (°C)

まわりの音の大きさ

> 入力

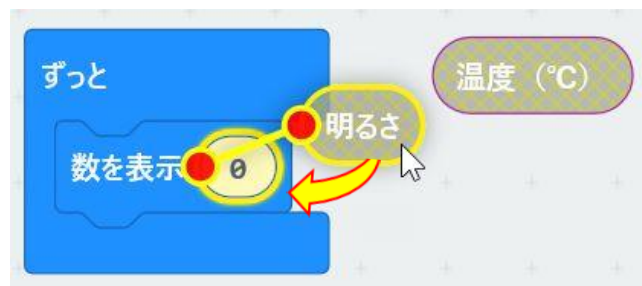
…その他

磁力 (μT)

x ▼

傾斜 (°)

ピッチ ▼



数を表示ブロックに  
センサーブロックを入れる

## センサーの値を 棒グラフ にしてみよう

ツールボックス

> L E D

棒グラフを表示する 値 0

最大値 0

センサーブロックを入れます

表示する値の最大値を入れます

※「明るさ」は、0～255

※「方角」は、 0～359 の範囲で変化します



手でLEDをおおうなどして  
グラフが変化するのを確認しよう

「最大値」はセンサーに合わせて  
変えることができます。

センサーの値を 音 にすることもできるよ

## 明るさを 音 で聴こう



micro:bitの明るさセンサーの値は、0～255の範囲で変化します。

人間に聞こえやすい音の周波数は、500～2000Hzの範囲です。

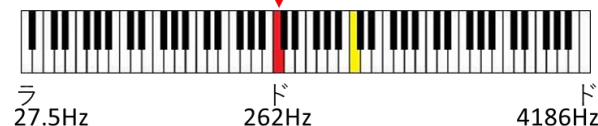
そこで計算ブロックを使って聞こえやすい範囲（4倍）にして音を鳴らしています。



音の高さ = 周波数

(1秒間に振動する数)

- ・ 1 秒間に約50回振動 = 50 Hz(ヘルツ)
- ・ ピアノの ドの音 = 262 Hz(ヘルツ)



## 磁力を 音 で聴こう






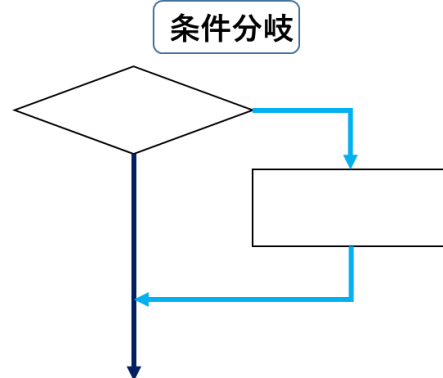
磁力（絶対値）は何倍にすると聞こえやすい音になるか、倍数を代えて実験してみよう。

# Step 4

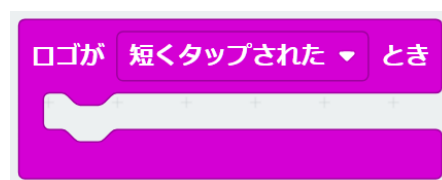
## ボタン A・B / ログをタップ(タッチ)

条件によって、動きが異なるプログラムを作ります。

- > ボタン A を押すと、 する。
- > ボタン B を押すと、 する。
- > ログをタップすると、 する。



ツールボックス  
> 入力



プログラム例)



チャレンジ

### 動きセンサーをつかおう [加速度計]

いろいろな組み合わせを試してみよう



左に傾けた時

アイコンを表示

画面が上の時

数字を表示



計算

乱数ブロック

0 から 10 までの乱数

乱数

サイコロをふったときに出る目のように、規則性がなく出る数のこと

ゆさぶられた 動き





6種類の選択課題を用意しました。  
自由を選んで取り組みます。

50

☆☆☆

Step-5 **Color <sup>カラー</sup> LEDを光らせよう** 単色/虹色/方角

☆☆☆

Step-6 **コンパス(方位磁針)をつくろう** ～磁気センサー～

☆☆☆  
★★☆

Step-7 **サイコロをつくろう** その1/その2

☆☆☆

Step-8 **じゃんけんゲームをつくろう** ～加速度センサー～

☆☆☆  
★★☆

Step-8 **磁力計でお宝をさがせ** グラフ/音 ～磁気センサー～

★★★

Step-9 **歩行者信号付き信号機をつくろう**

時間内に終わらない時は、自宅で行ってみよう！

## Step 5

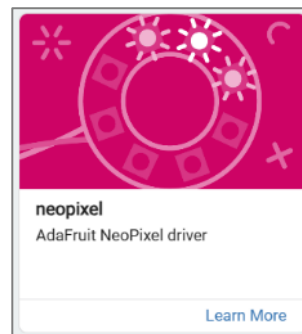
# Color LEDを光らせよう

☆☆☆

n:bitにはカラーLEDが付いています。  
3色のLEDの組み合わせで約1677万色（ $256 \times 256 \times 256$ ）を表現できます

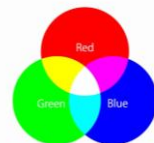
拡張機能から、  
neopixelを追加

+ 拡張機能



Color LED  
NeoPixel

光の三原色



Red / Green / Blue  
(赤) (緑) (青)



ColorLEDキャップ

計算

Neopixel

最初だけ

変数 **strip** を 端子 **P1** に接続しているLED **1** 個のNeoPixel (モード **RGB (GRB順)**) にする

ずっと

**strip** を **赤** 色に点灯する 赤色が点灯

- ✓ 赤
- だいだい
- 黄
- 緑
- 青
- あい
- すみれ
- 紫
- 白
- black

10色が選べます

**Black (黒)** は  
どのように光る ???

赤色の以外も選んで光らせてみよう！！

◆LEDが明るすぎる場合は調整しよう **Point!**

Neopixel  
> その他

**strip** の明るさを **255** に設定する

# RGB LEDを虹色に光らせよう！

**最初だけ**

変数 **strip** を 端子 **P1** に接続しているLED **24** 個のNeoPixel (モード **RGB (GRB順)**) にする

**strip** をレインボーパターン (色相 **1** から **360**) に点灯する

**ずっと**

**strip** に設定されている色をLED **1** 個分ずらす (ひとまわり)

**strip** を設定した色で点灯する



チャレンジ



## 方角でRGB LEDの色を変化させよう！

使うブロック

Neopixel hue saturation luminosity

色相 彩度 明度

**ずっと**

**strip** を hue 方角 (°) saturation **100** luminosity **50** 色に点灯する

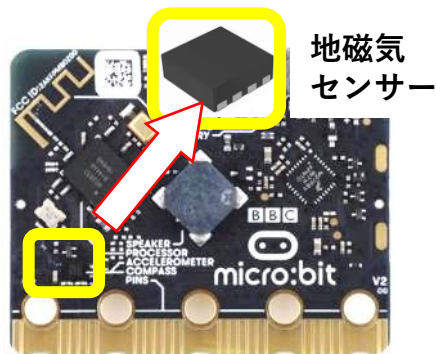
色を角度で表す方法「色相」があります。その角度に方角 (°) ブロックを使い、n:bitの方角でLEDの色を変化させます。

# Step 6

## コンパス(方位磁針)をつくろう

★★☆

条件分岐

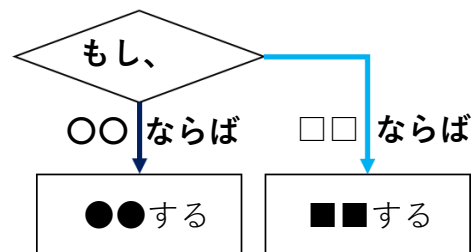


地磁気  
センサー

### 地球は大きな磁石 (地磁気)

micro:bitの地磁気センサーは、  
方位を方角 (0~359°) で表します。

[条件分岐] のしくみを使い、  
「方角」によって方位を表示するプログラムをつくります。



### 使うブロック

基本

文字列を表示

"Hello!"

入力

方角 (°)

論理

もし 真 なら

でなければ

+

ずっと

もし 方角 (°) < 45 なら

文字列を表示 N

でなければもし 方角 (°) < 135 なら

文字列を表示 E

でなければもし 方角 (°) < 225 なら

文字列を表示 S

でなければもし 方角 (°) < 315 なら

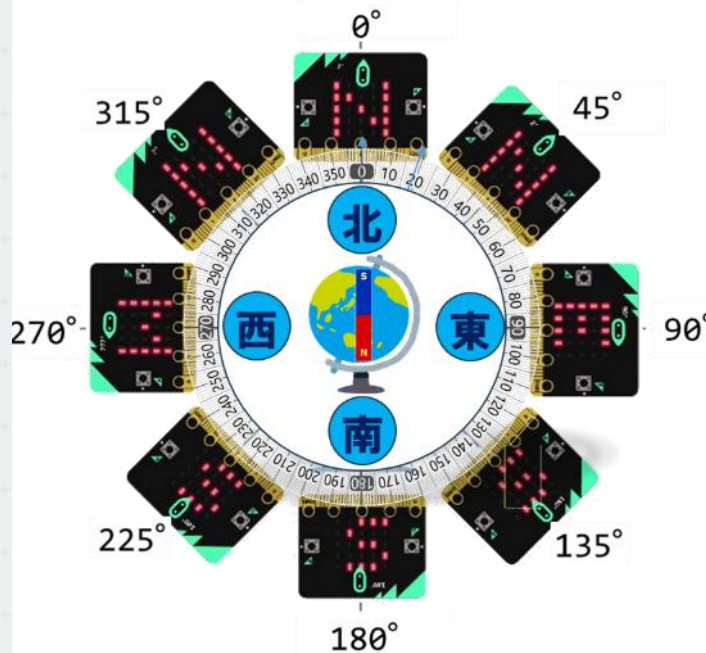
文字列を表示 W

でなければ

文字列を表示 N

← Nは45° とここで2回表示しています。なぜ？

方位 (東西南北) と  
方角 (360°) の関係



East : 東  
West : 西

North : 北  
South : 南





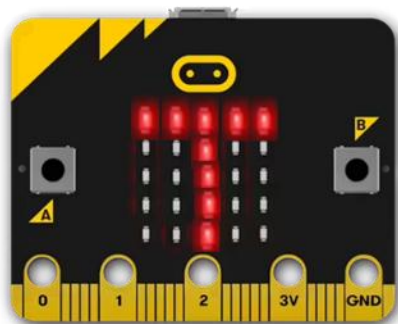
## 磁気センサーの校正 (キャリブレーション)

※構成に失敗した時は、いったん電源を切って  
サイド電源を入れると校正が始まります。

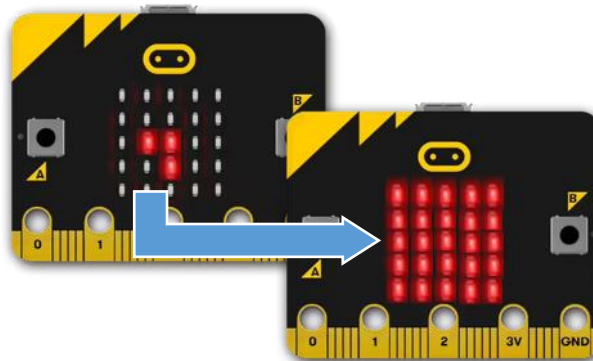
センサーの校正  
センサーが正しく動作  
するよう誤差を調整  
することです

(傾けて画面をうめる)

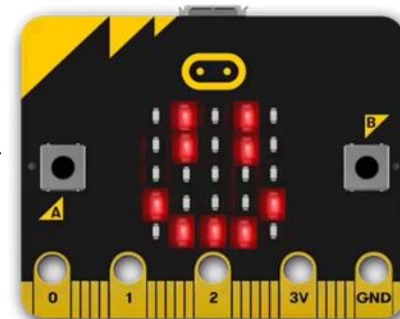
① **TILT TO FILL SCREEN** と  
LEDにスクロール表示されます。



② 中央部分のLEDが数個点灯するの  
で、micro:bitを左・右・上・下い  
ろいろな方向に傾けて、25個  
すべてのLEDを点灯させます。



③ 校正が終わりLEDに笑顔が点  
灯します。



## 真北と磁北は、同じ？違う？

方位磁石は、地球全体が磁石の性質（地磁気）を持つことを  
利用して、磁石の針で南北の方角を示します。

2020年現在、札幌では地図の北よりも約10度西にずれますが、  
那覇ではそのずれは約6度と、場所によってずれる幅が異なり  
ます。



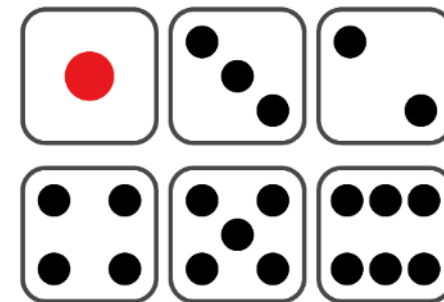
国土地理院による「偏角」の解説イメージ図  
(出典：国土地理院ウェブサイト)



## Step 7 サイコロをつくろう！ その1

★☆☆

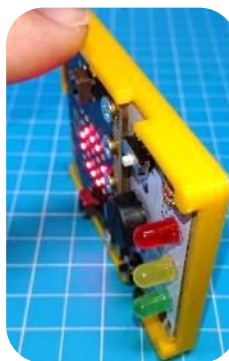
micro:bitの画面が上になると、  
ランダムな数字が表示するようにプログラムします。



ツールボックス  
→ 計算

0 から 10 までの乱数

サイコロの数字が出るように値を変えよう  
数値は、？から？までにするといいかな？？？



・画面が上になった時、メロディも鳴るようにしてみよう！

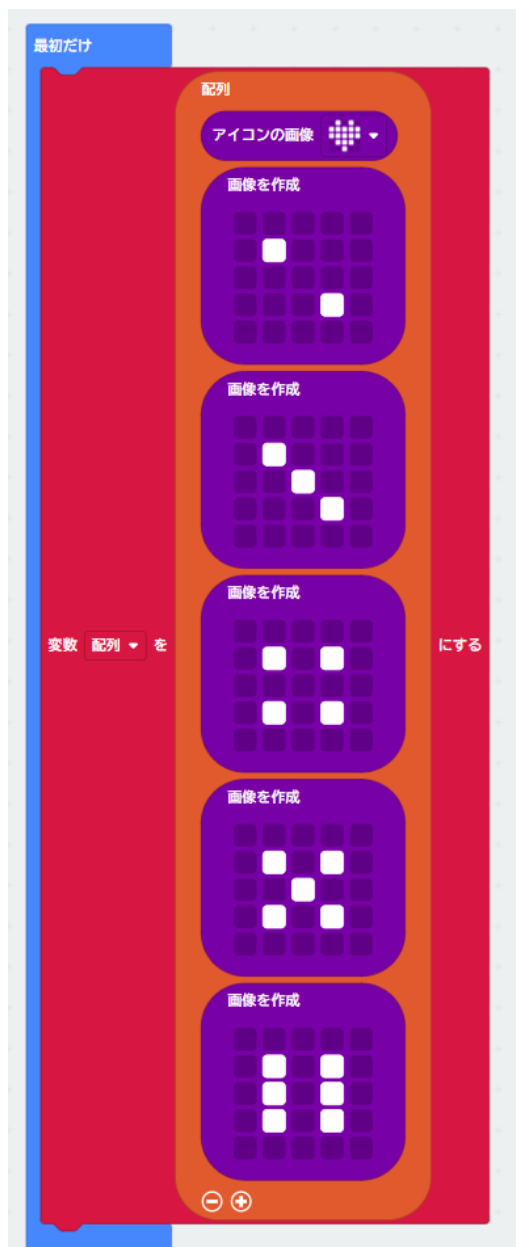
メロディ  をテンポ 120 (bpm)で演奏する



# サイコロをつくろう！ その2


★★☆

サイコロの目（画像）で表示する（画像）  
Aボタンを押すと、サイコロが振られます。



このプログラムでは「**配列**」**ブロック**を使います。  
ちょっと難しいので、よくわからない方はインターネットで  
「配列とは」と、検索してみてください。  
やさしい解説がたくさんでてくると思います。

さらに今回のプログラムでは、配列を数字や文字でなく  
「画像」を使うというテクニックを使います。  
つまり、サイコロの目を画像として配列にすることで、  
とても短いプログラムを作ることができます。

ツールボックスにある「画像」のブロック一覧を見ると、  
アイコンも画像として扱えます。  
このプログラムではサイコロの1の目を  アイコンにし  
てます。

## Step 8

# じゃんけんゲームをつくろう！ その1

<https://nextday-kids.com/wp/1710/>

☆☆☆

### Mission

micro:bitじゃんけん  
をつくろう！



## ●じゃんけんゲームを設計する

micor:bitでどのように動かす（表示する）かを考えます。  
まず、「ぐー」「ちょき」「ぱー」の形の代わりに、  
数字の「1」「2」「3」を表示させます。  
ルールは、数字の大きい方を勝ちにして、人が判断します。

「さいしょはグー」

・ボタンを押してリセット

「じゃら、けつ、ほい」

・ふってじゃんけん

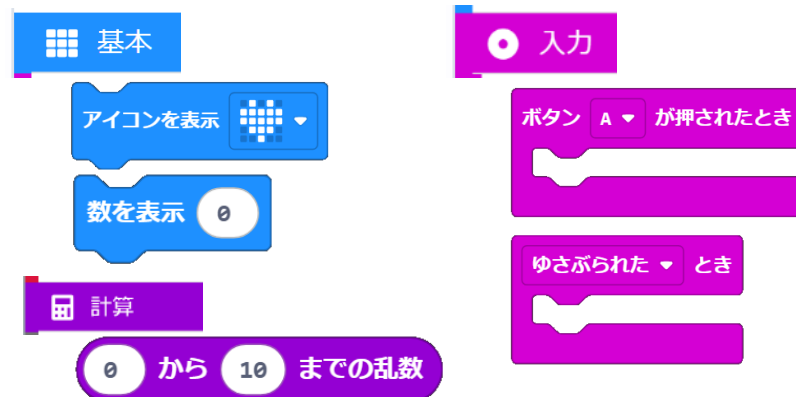
・1/2/3の数字を表示

・数字の多い方が勝ち

- ・プログラムが起動したことがわかるように、最初に❤️ハートマークを表示します。
- ・じゃんけんの準備はAボタンとBボタンを両方押して、知らせます。今回は、数字の0を表示します。
- ・micro:bitを振ると、数字の1, 2, 3のいずれかを表示します。

## ◆プログラムを作成する

### 使うブロック



## ♠改良しよう

思い通りにプログラムが動いたならば、ぜひ、改良を考えてみよう。

たとえば、  
「ゆさぶられたとき」や  
「ボタンA+Bが押されたとき」が、  
ハッキリと分かるように、音を鳴らす。



# じゃんけんゲームをつくろう！ その2

☆☆☆

## ●じゃんけんゲームを設計を一部変更する

- ・micro:bitを振ると、  
数字の1, 2, 3のいずれかに対応した  
アイコンを表示します。

変数は、「数を記憶するための箱」です。



### ① じゃんけん用変数ブロックをつくる



変数の名前  
**JANKEN**

変数ブロック



### ② グー・チョキ・パーのアイコンをつくる



## 自分でデザインしてみよう！

値	アイコン
1	: グー
2	: チョキ
3	: パー

1 : グー



2 : チョキ



3 : パー



改良しよう

その1と同じく音（メロディ）が  
鳴るようにしてみよう

最初だけ

アイコンを表示



変数 JANKEN を 0 にする

ボタン A+B が押されたとき

変数 JANKEN を 0 にする

数を表示 JANKEN

ゆさぶられたとき

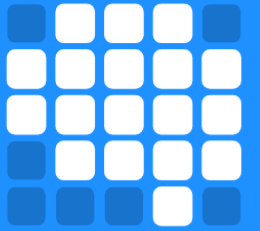
変数 JANKEN を 1 から 3 までの乱数 にする

ずっと

1 ⇒ グー

もし JANKEN = 1 なら

LED画面に表示

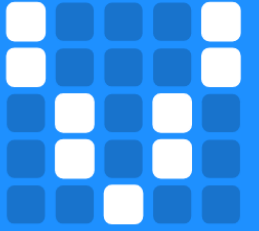


ずっと

2 ⇒ チョキ

もし JANKEN = 2 なら

LED画面に表示

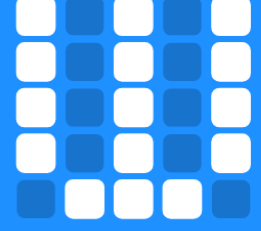


ずっと

3 ⇒ パー

もし JANKEN = 3 なら

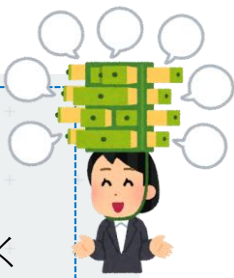
LED画面に表示



変数 = 「変わる数」

変数にはわかりやすい「名前」を  
つけることができます。

[変数]ブロックは、micro:bit内につく  
られる「引き出し」です。  
micro:bitの電気が止まると内容は消え  
てしまいます。





## Step 9

# 磁力計でお宝をさがせ！

micro:bit + n:bit を使って、隠れているお宝（磁石）を探す、磁力計プログラムをつくります。

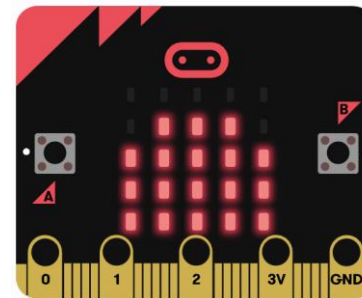


## プログラム設計

磁力そのものは、見たり聞いたりすることはできません。そこで、磁力を音や光に変えるプログラムを作り、磁石の存在が人にも分かるしくみを考えます。

nextday-kidsのホームページでは、6種類の方法を紹介しています。今回はその中から2種類を取り上げます

- ① 音の高低で探す？
- ② 赤色LEDの明るさで探す？
- ③ 棒グラフで探す？
- ④ Color LEDの明るさで探す！（n:bit II用）
- ⑤ 音の高低で探す！（v2用）
- ⑥ 画像の明るさと動きで探す！



棒グラフで探す

磁力が  
強い＝バーが多い  
弱い＝バーが少ない

☆☆☆

## ③ 棒グラフで探す？

磁力の強弱を、micro:bitのLED画面に棒グラフ表示して、磁石を探知するプログラムです。

## 使うブロック



## 使うブロック

## 変数

変数 A ga OSARETA を 0 にする

A ga OSARETA

↑変数 A ga OSARETA をつくる

## 入力

ボタン A が押されたとき

## 論理

もし 真 なら

+

真

偽

micro:bitV2に新しく内蔵されたスピーカーを鳴らして磁石を探知します。音を鳴らしたり止めたりできる「Aボタンで音を鳴らす」「Bボタンで音を止める」という機能も加えました。変数[Aボタンが押された]に真偽値を入れて、判断条件としています。Aボタンを押した時は、真という値が変数に入り、Bボタンを押したときは、偽という値が入ります。真偽の計算のことをブール代数とも呼びます。

ずっと

もし A ga OSARETA なら

音を鳴らす 高さ (Hz) 磁力 (μT) z

でなければ -

すべての音を停止する

+

ボタン A が押されたとき

変数 A ga OSARETA を 真 にする

ボタン B が押されたとき

変数 A ga OSARETA を 偽 にする

# Step 10

## 歩行者信号付き信号機をつくろう！

★★★

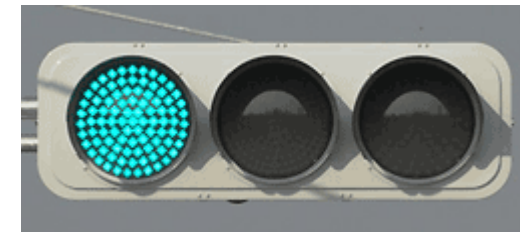
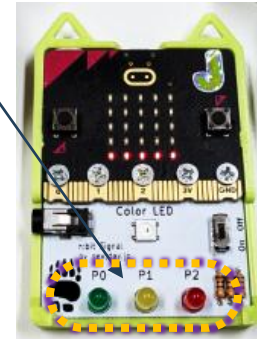
n:bitの3色LEDは車用、micro:bitのLEDは歩行者用の信号にします。

### プログラム設計

- ・最初は、車用は緑「●」、歩行者用は「止まれ」を表示させます。
- ・ボタンAが押されたなら、まず、車用を黄「●」にします。
- ・2秒後に、車用を赤「●」にします。
- ・さらに2秒後に、歩行者用を「すすめ」の表示にします。
- ・5秒後に歩行者用の「すすめ」表示を6回点滅させます。
- ・歩行者用を「止まれ」の表示にします。
- ・1秒後に車用を緑「●」にします。

#### 3色 LED

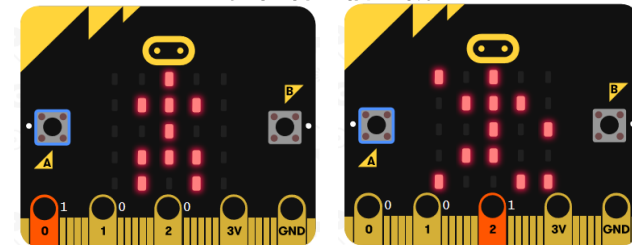
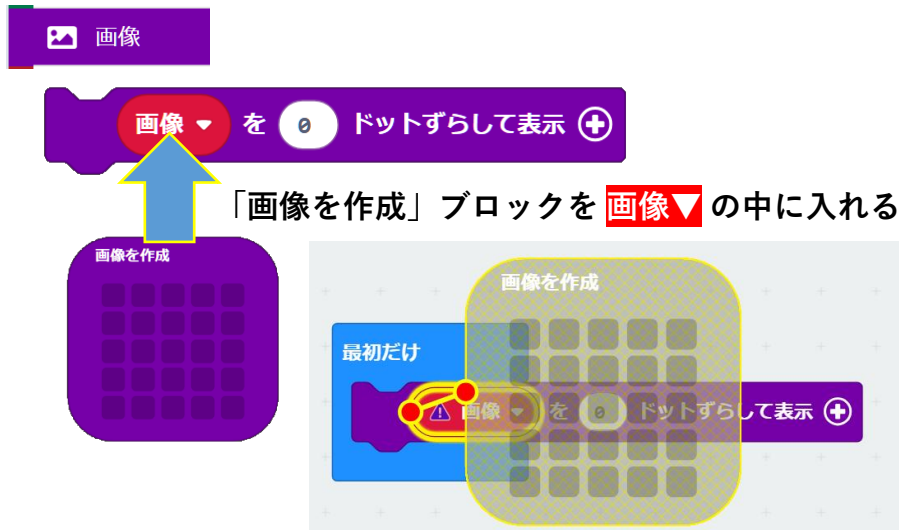
端子	色
P0	緑
P1	黄
P2	赤



車用信号機

歩行者用信号機

### ① 歩行者用画像ブロック「止まれ」をつくる



n:bitの3色LEDを点滅させるには、  
それぞれの端子に電気を「流す」「止める」が必要です。  
入出力端子ブロックの「**デジタルで出力する端子**」を使います。

高度なブロック

◎ 入出力端子

デジタルで出力する 端子 P0 ▼ 値 0

デジタルで出力する 端子 P0 ▼ 値 0

端子	色
P0	緑
P1	黄
P2	赤

端子色

P0：緑

P1：黄

P2：赤

P0 P1 P2 P3

P4 P5 P6 P7

P8 P9 P10 P11

P12 P13 P14 P15

P16 P19 P20

デジタルで出力する 端子 P0 ▼ 値 0

値 状態

0：消える  
電気を「止める」

1：点く  
電気を「流す」

Value 0

Value 1



ボタンA を押したら 黄 が点灯

ボタンB を押したら 赤 が点灯

ボタン A ▼ が押されたとき

ボタン A が押されたとき

デジタルで出力する 端子 P0 値 0 緑 (P0) が消える

デジタルで出力する 端子 P1 値 1 黄 (P1) が点く

一時停止 (ミリ秒) 2000

デジタルで出力する 端子 P1 値 0 黄 (P1) が消える

デジタルで出力する 端子 P2 値 1 赤 (P2) が点く

一時停止 (ミリ秒) 2000

大きな画像を作成



を 1 ドットずつ 200 ミリ秒ごとにスクロール

一時停止 (ミリ秒) 5000

くりかえし 6 回

明るさを設定する 0

一時停止 (ミリ秒) 200

明るさを設定する 255

一時停止 (ミリ秒) 500

歩行者用信号アイコン 歩く が  
6 回点滅する

画像を作成



を 0 ドットずらして表示

一時停止 (ミリ秒) 1000

デジタルで出力する 端子 P2 値 0 赤 (P2) が消える

デジタルで出力する 端子 P0 値 1 緑 (P0) が点く

最初だけ

デジタルで出力する 端子 P0 値 1

画像を作成



を 0 ドットずらして表示

画像

画像 を 1 ドットずつ 200 ミリ秒ごとにスクロール

大きな画像を作成



ループ

くりかえし 4 回

LED

その他

LED画面の明るさを設定する 255



プログラミングで  
あ・そ・ぼ

micro:bit

ライトレース  
ロボットカー  
を走らせよう!



n:bit II

8連RGB LED

※3.5mmオーディオケーブルで  
micro:bitと接続

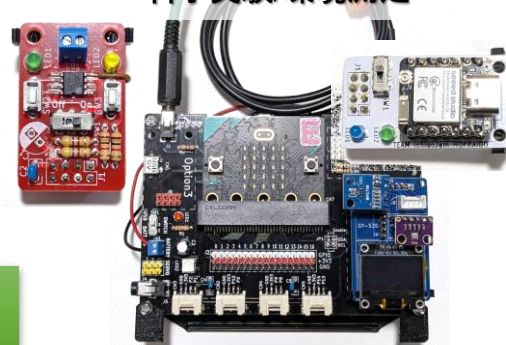


透明アクリル板やペットボトルキャップ  
をキャンバスにしてイラストを描く。



SCIENCE Palette

科学実験/環境測定



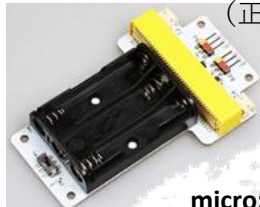
作って・測って・観察して・考える

micro:bit v2 市販教材の購入 スイッチサイエンス

(正規代理店)



Maqueen Lite V5

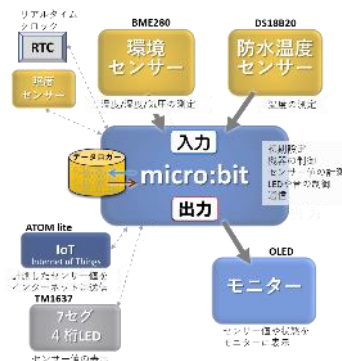


micro:bit用  
ワークショップモジュール

nextday プログラミングキット

講座内容について詳しい解説や教材の説明、  
プログラミング講座や出展等を開催案内を  
紹介しています。

<https://nextday-kids.com/>



データサイエンス入門教材

micro:bitと環境測定ユニット  
「お天気CUBE」をプログラミ  
ングして、温度・湿度・気圧を  
測定してデータ化。データを可  
視化して気候変動等の課題解決  
について考えます。

いまをつくる!



NPO法人 NEXTDAY は  
子供たちの学びを支援しています

お問い合わせは nextday@ict.skr.jp

未来を創る!

お問い合わせは nextday@ict.skr.jp

