

.PrimaryKeying

あ・そ・ほ

で

未来を創る！

NPO法人 NEXTDAY

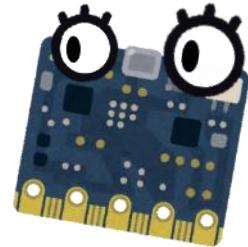
n:bit



& micro:bit



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

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

はじめてのmicro:bit v2

全員で同じ内容
を取り組みます。

- 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)

45

n:bit
& **micro:bit**

選択課題から
自由に選んで
取り組みます。



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

50

- 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

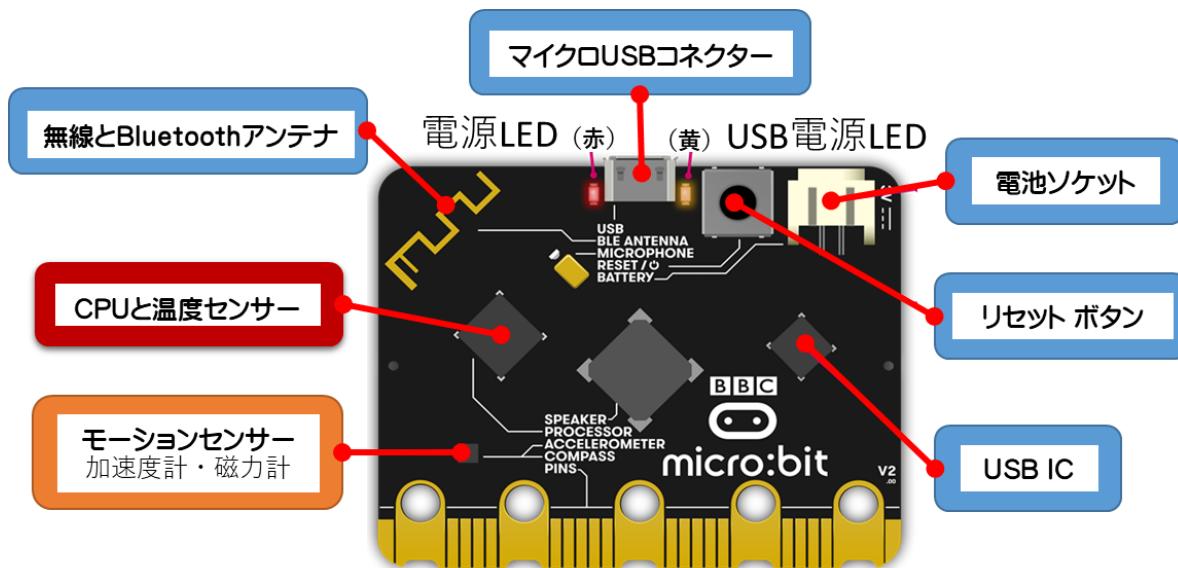
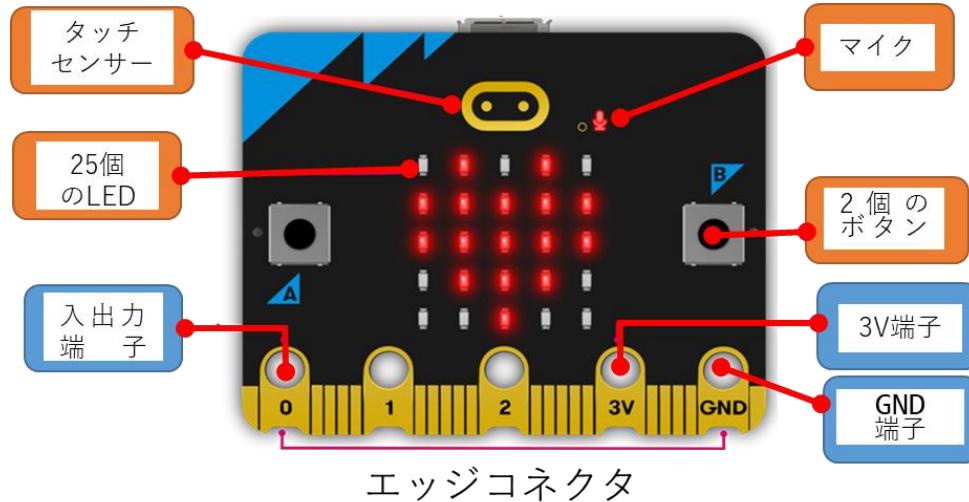
- Step-8 **じゃんけんゲームをつくろう** ～加速度センサー～

- 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**

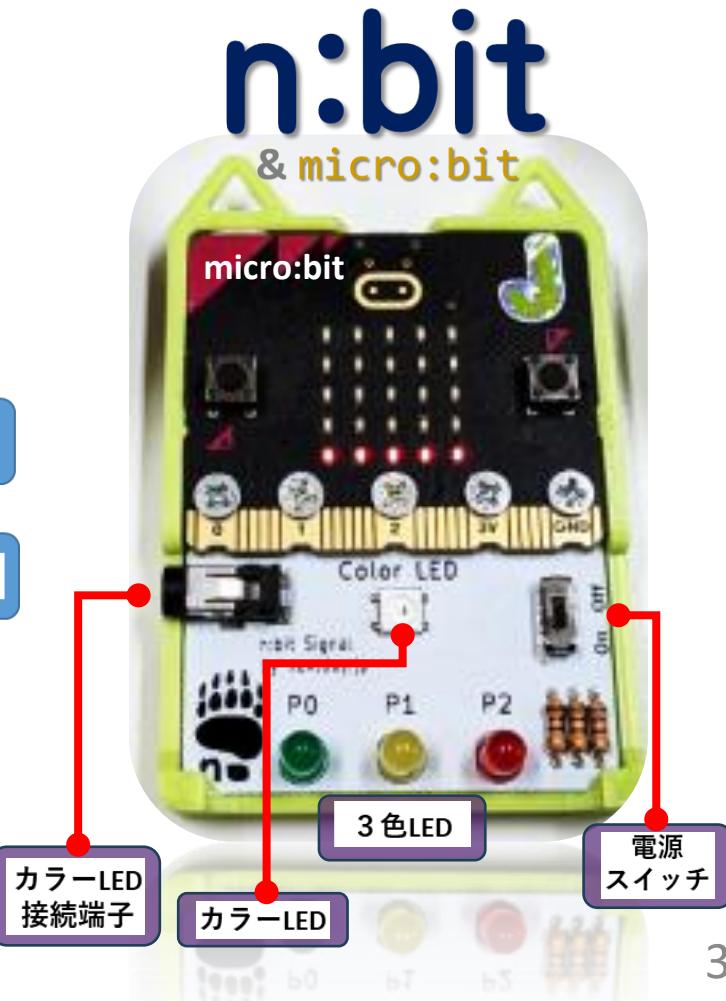
2

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



詳しくは

<https://microbit.org/ja/guide/features/>

はじめてのmicro:bit

・オンライン版makecode (無料)

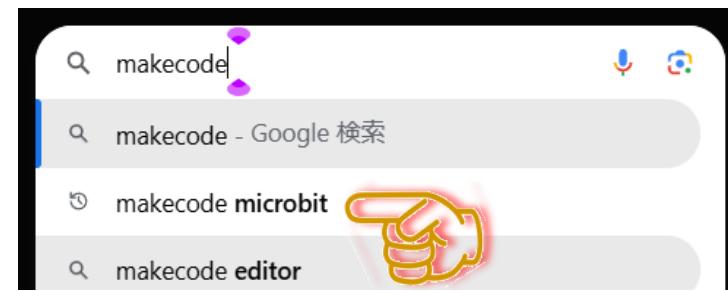
インターネットに接続しているパソコンのブラウザ
(Microsoft Edge または Google Chrome を推奨) でが使えます。

<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

→ Makecode画面で
[F11：全画面]モードにする

2. クリックロック防止

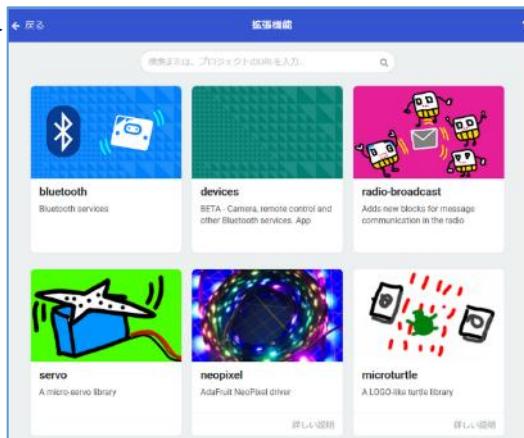
→ マウスのプロパティでクリックロック機能解除







・拡張機能



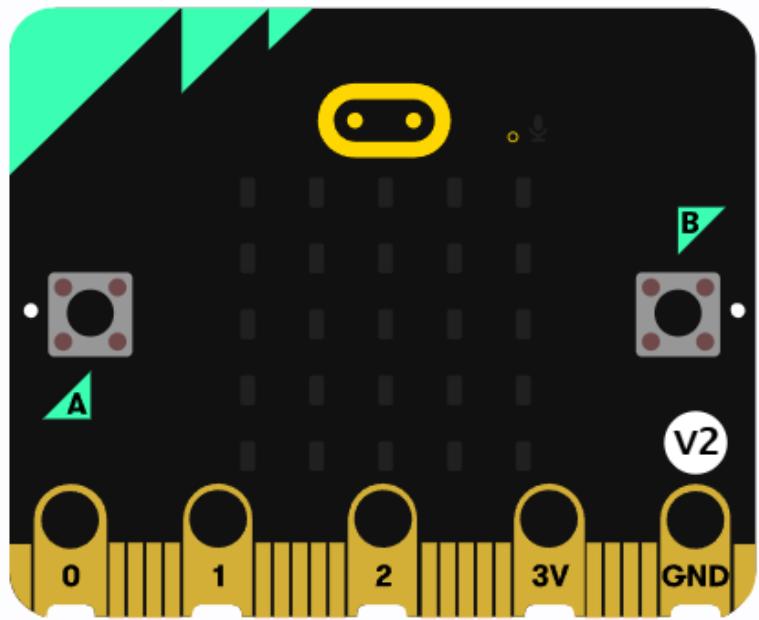
より複雑な制御や機能を実現するための追加ブロックです。複数の機能を組み合わせて

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。

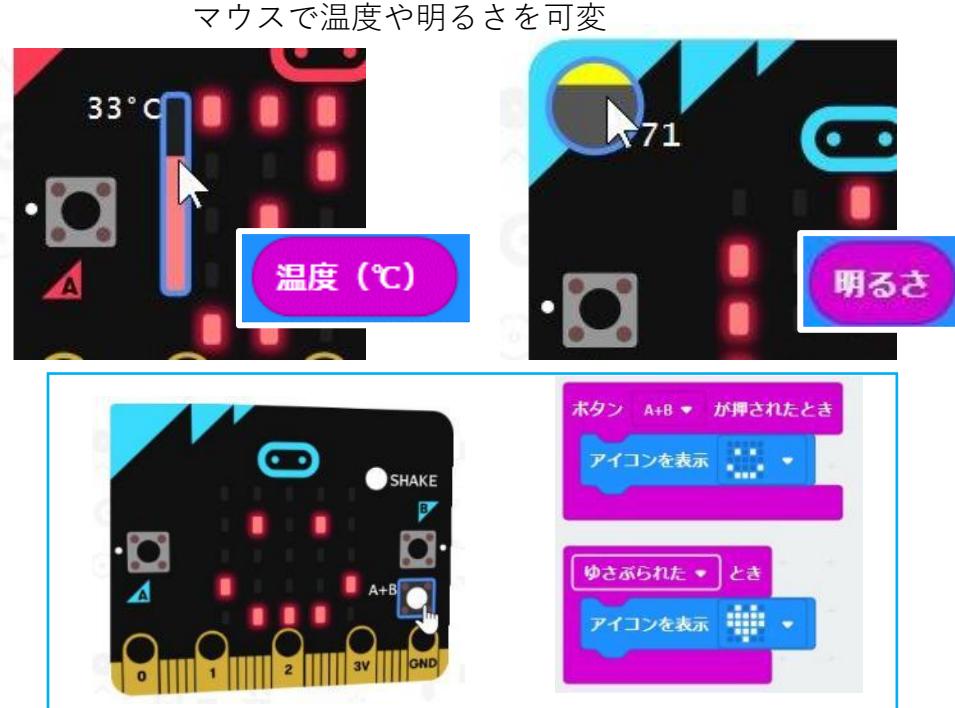


はじめてのmicro:bit

・シミュレーター



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。

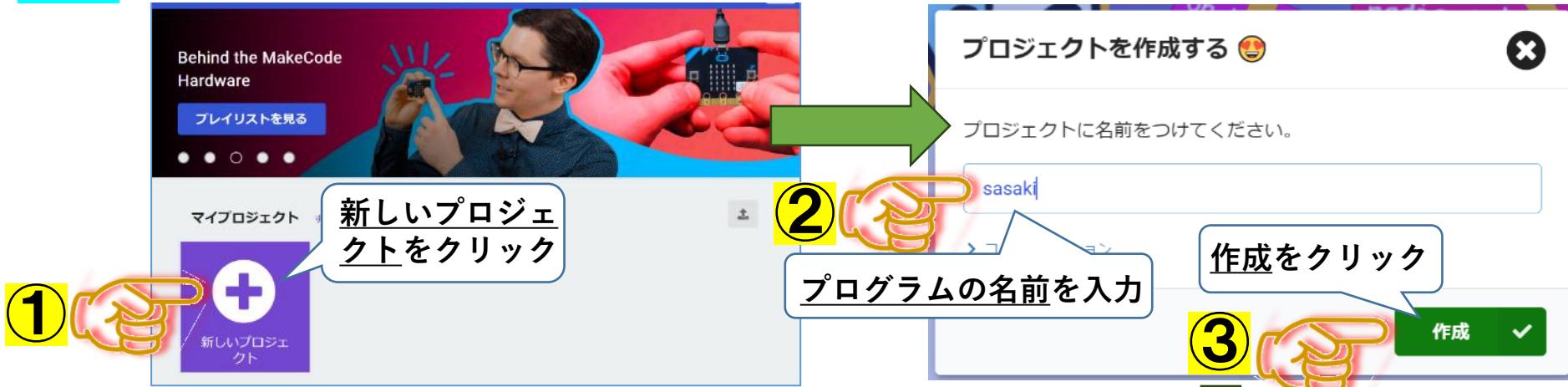


↑マウスでボタンを押したり
ゆさぶることができる

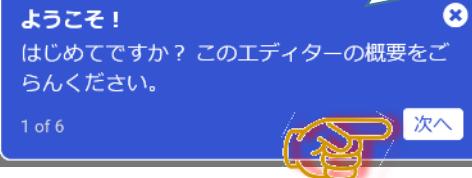


S1_1

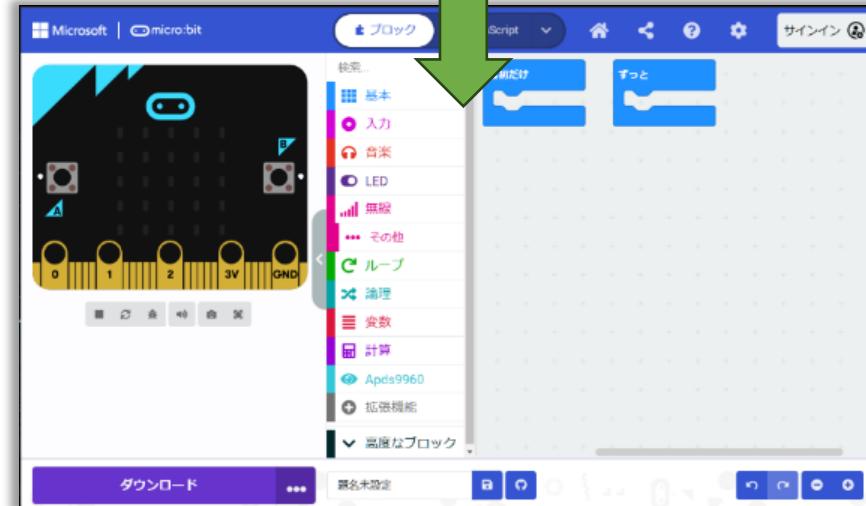
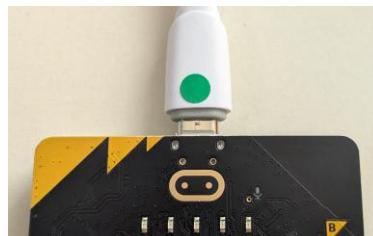
Microsoft MakeCode for micro:bit で新しいプロジェクトをつくる



もし、“ようこそ！”のポップアップが出たら 次へ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/>

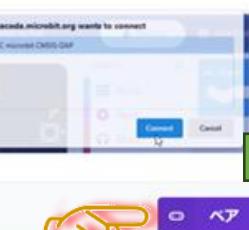


② ダウンロード 

ロゴが表示されている場合は、接続済みです。
このペアリングは不要です。

③ 2. micro:bit をブラウザにペアリングしてください。

下のペアボタンを押してください
ブラウザの上部にウィンドウが表示されます。
micro:bit を選択し、接続を押してください。



④ 選択 

makecode.microbit.org が接続を要求しています

BBC micro:bit CMSIS-DAP

接続

キャンセル

⑤ micro:bit に接続しています

あなたのmicro:bit が接続されました！「ダウンロード」を押すと自動的にコードが micro:bit にコピーされます。



接続中を示すロゴを確認

⑥ ダウンロード 

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ら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

The image shows the Scratch interface. On the left is the 'Toolbox' with categories: 基本 (Basic), 入力 (Input), 音楽 (Music), LED, 無線 (Wireless), ループ (Loop), 論理 (Logic), 変数 (Variables), and 計算 (Math). The 'Basic' category is selected, highlighted with a yellow arrow. Inside the 'Basic' category, there are blocks for '数を表示' (Show Number), 'LED画面に表示' (Show on LED Screen), 'アイコンを表示' (Show Icon), and '文字列を表示' (Show String). The '文字列を表示' block is highlighted with a yellow arrow and has a hand cursor over it. On the right, the 'Control' category is shown with blocks for '最初だけ' (Once Only) and 'ずっと' (Forever). Below these are three specific blocks: '文字列を表示 "Hello!"' (Show String "Hello!"), '数を表示 0' (Show Number 0), and 'アイコンを表示' (Show Icon). A callout box points to the '最初だけ' block with the text '・最初だけ' (Once Only). Another callout box points to the 'ずっと' block with the text '・何度も同じ動作を繰り返す' (Repeat the same action many times). At the bottom, there are three flowchart diagrams: '順次' (Sequential) showing a vertical flow from top to bottom; '繰り返し' (Loop) showing a vertical flow with a loop returning to the top; and '最初だけ' (Once Only) showing a vertical flow with a single block.

検索...

■ 基本

… その他

○ 入力

◐ 音楽

◐ LED

◐ 無線

◐ ループ

◑ 論理

☰ 変数

▤ 計算

■ 基本

数を表示 0

LED画面に表示

アイコンを表示

文字列を表示 "Hello!"

最初だけ

ずっと

順 次

繰り返し

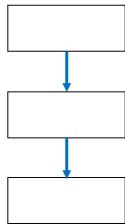
最初だけ

・最初だけ

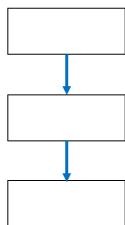
・何度も同じ動作を繰り返す

11

順 次



順 次



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

鳴らす メロディを鳴らす  テンポ 120 (bpm) 終わるまで ▾

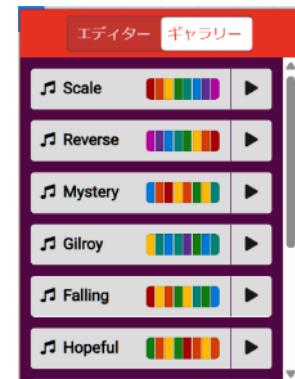
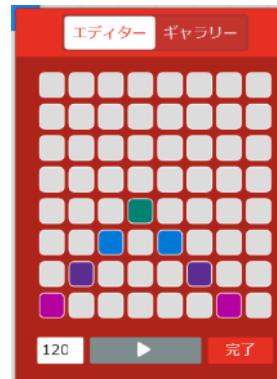
鳴らす 音を鳴らす 高さ (Hz) 真ん中のド 長さ 1 ▾ 拍 終わるまで ▾

音を鳴らす 高さ (Hz) 真ん中のド

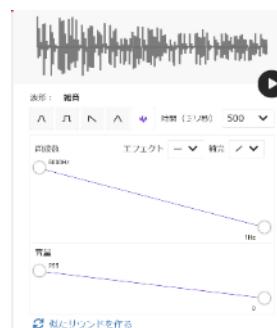
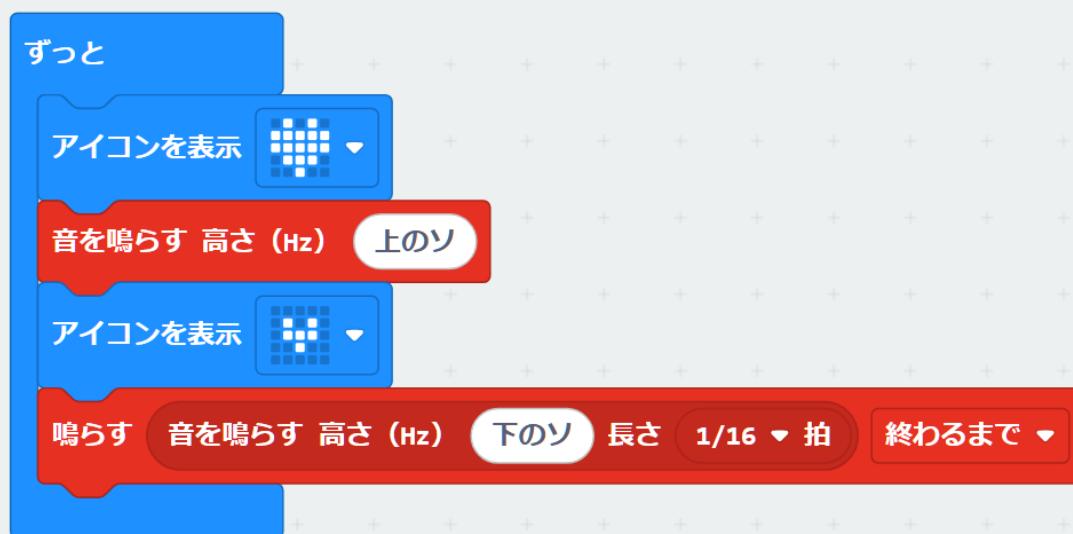
鳴らす くすくす笑う ▾ 終わるまで ▾

鳴らす  + 終わるまで ▾

自分で音や効果音をつくりたり、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

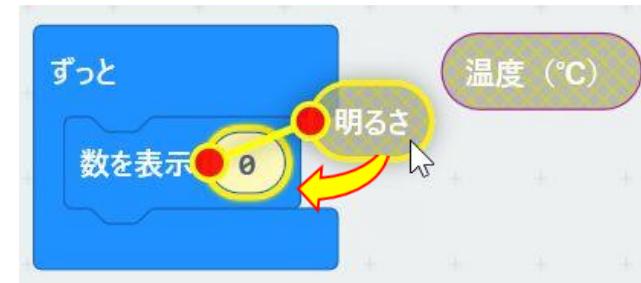
…その他

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

> LED

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

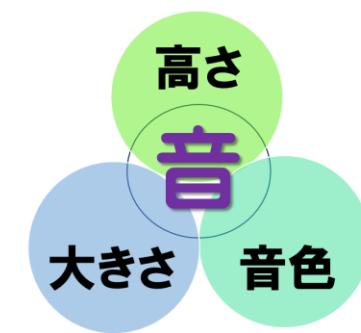
明るさを音で聴こう



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。

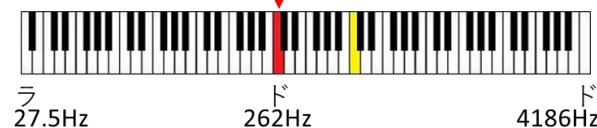
そこで計算ブロックを使って聞こえやすい範囲（4倍）にして音を鳴らしています。



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



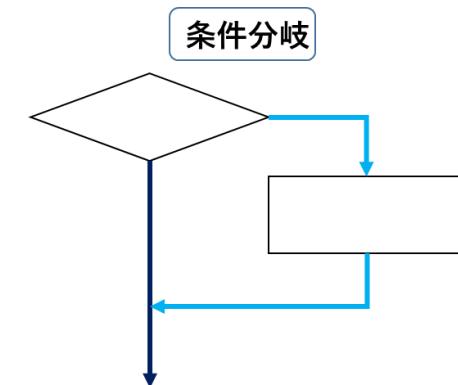
磁力（絶対値）は何倍にすると聞こえやすい音になるか、倍数を代えて実験してみよう。

Step 4

ボタン A・B / ロゴをタップ(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。
- > ロゴをタップすると、  する。

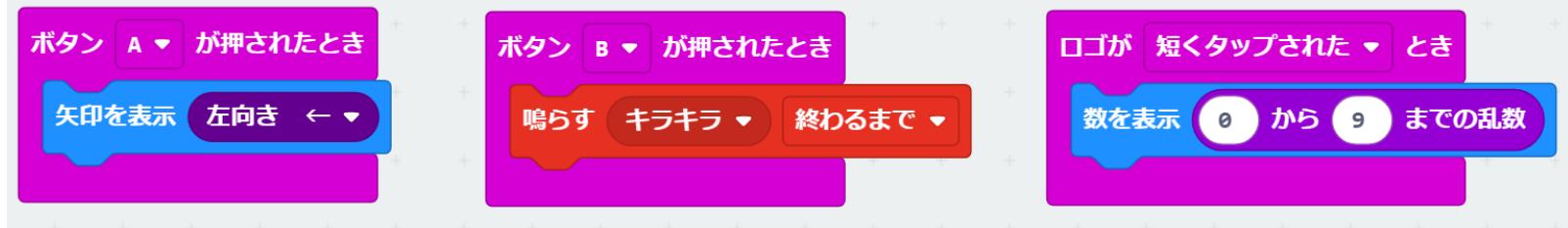


ツールボックス

> 入力



プログラム例)



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

ゆさぶられた ▾ とき

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。
自由に選んで取り組みます。

50



Step-5 **Color LEDを光らせよう** カラー 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、自宅で取り組んでみよう！

Step 5

Color LEDを光らせよう

★★★

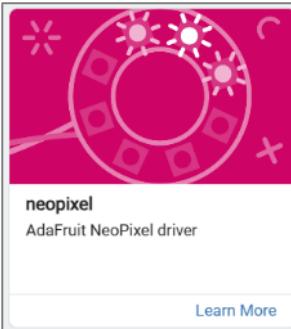
n:bitにはカラーLEDが付いています。

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

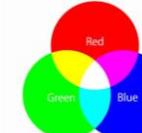
+

拡張機能

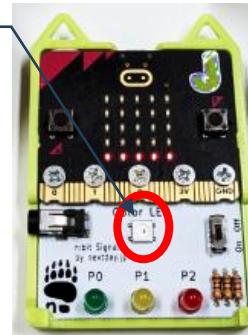


Color LED
NeoPixel

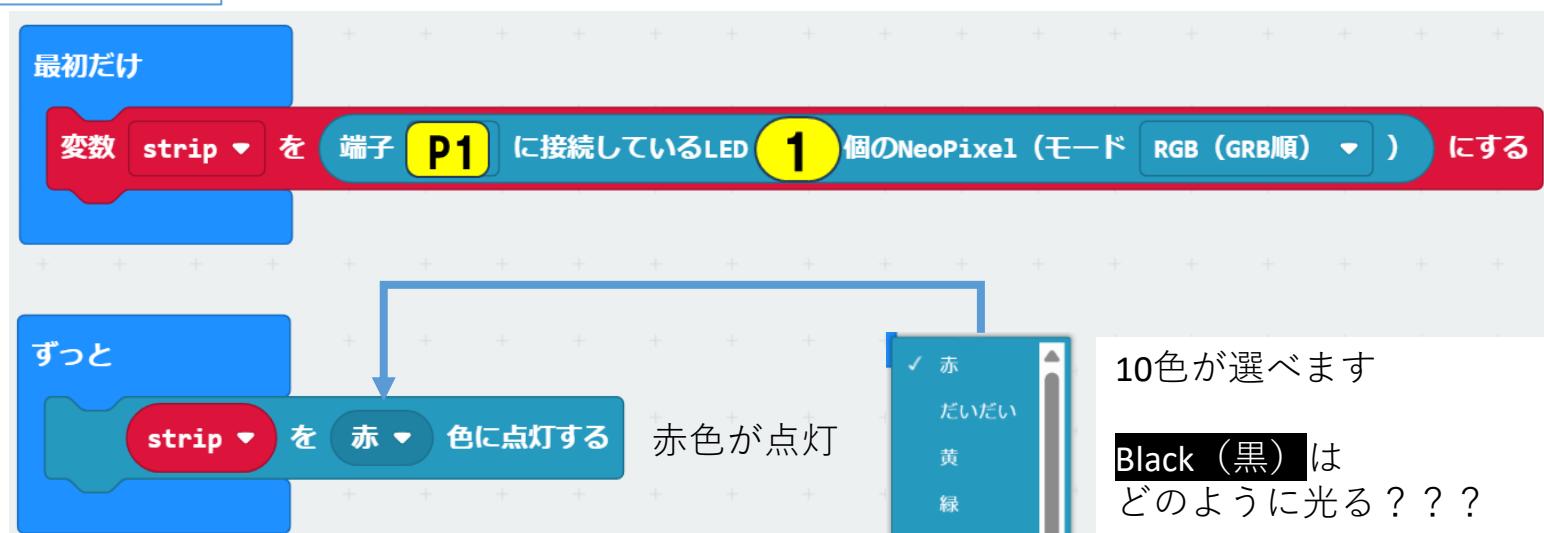
光の三原色



Red / Green / Blue
(赤) (緑) (青)



ColorLEDキャップ



赤色の以外も選んで光らせてみよう！！



10色が選べます

Black (黒) は
どのように光る???

◆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

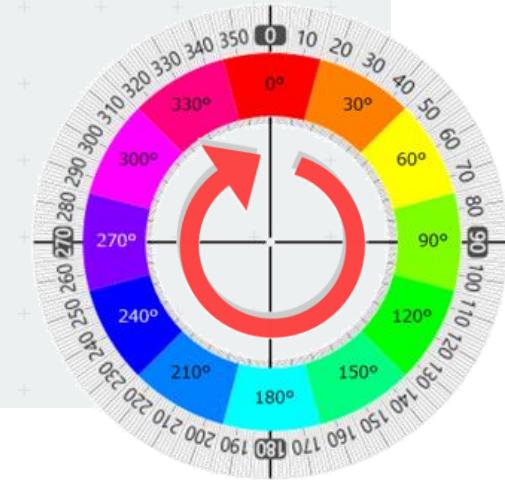
変数 **strip** を 端子 **P1** に接続しているLED **24** 個のNeoPixel (モード **RGB (GRB順)**) にする

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



方角でRGB LEDの色を変化させよう！

使うブロック

Neopixel hue **0** saturation **0** luminosity **0**

色相

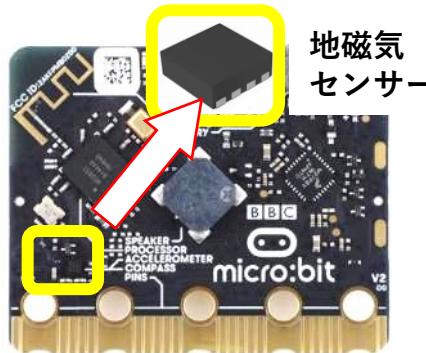
彩度

明度

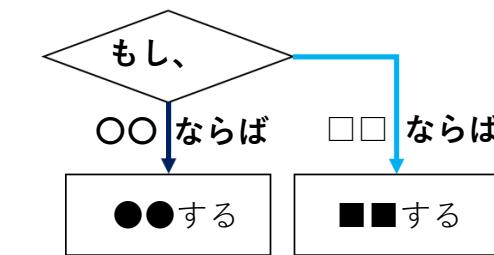
ずっと

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。

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 (°)

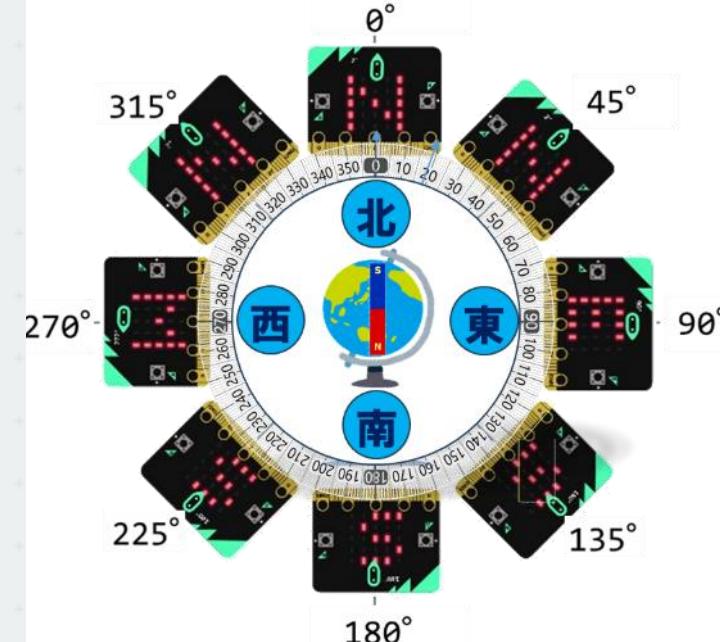
論理

もし 真 ▼ なら

でなければ □

□

0 □ 0

方位（東西南北）と
方角（360°）の関係

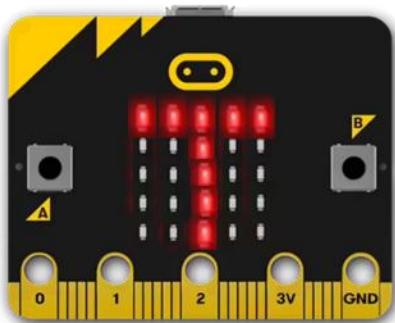
East	：東	North	：北
West	：西	South	：南



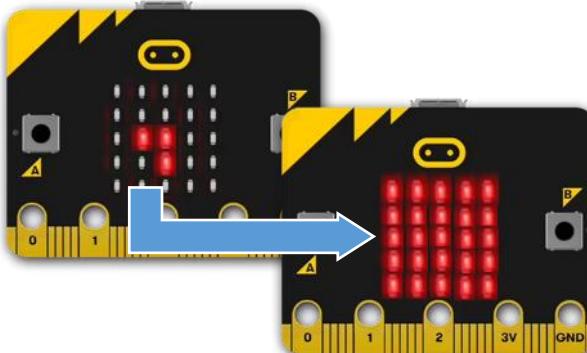
磁気センサーの校正 (キャリブレーション)

(傾けて画面をうめる)

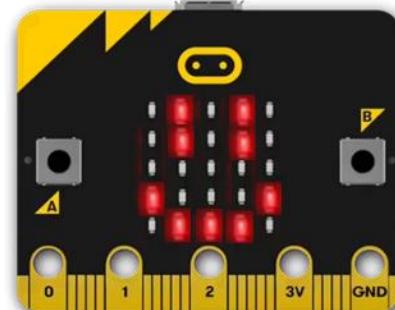
- ① **TILT TO FILL SCREEN** と LED にスクロール表示されます。



- 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。

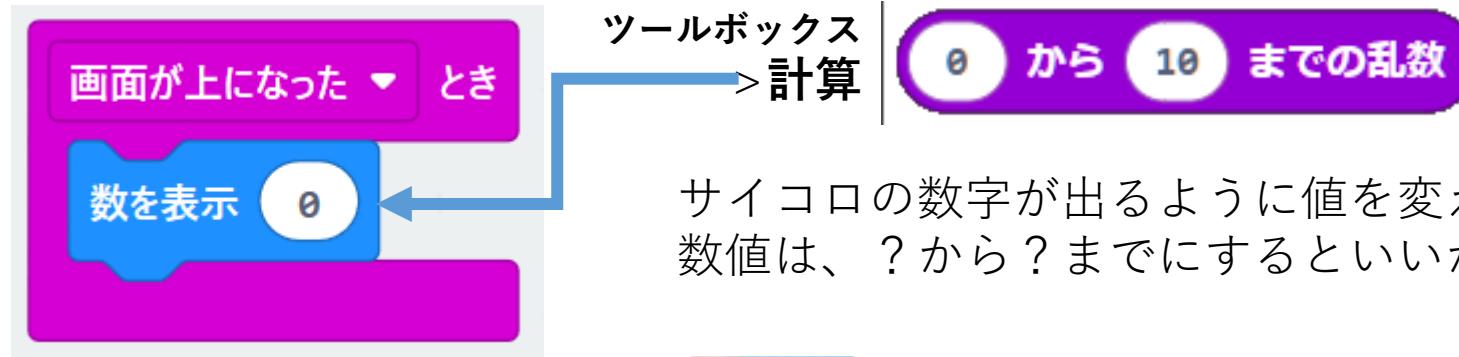
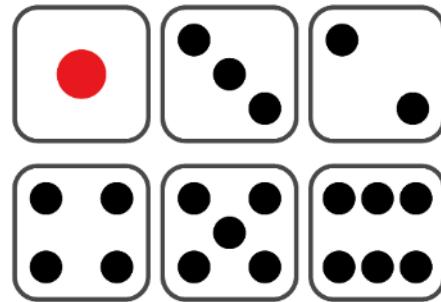


センサーの校正
センサーが正しく動作するよう誤差を調整することです

Step 7

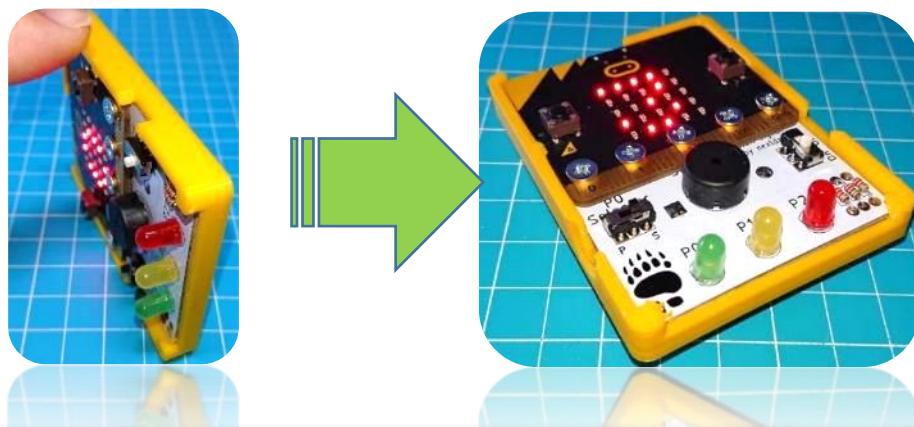
サイコロをつくろう！ その1 ★★★

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？？？



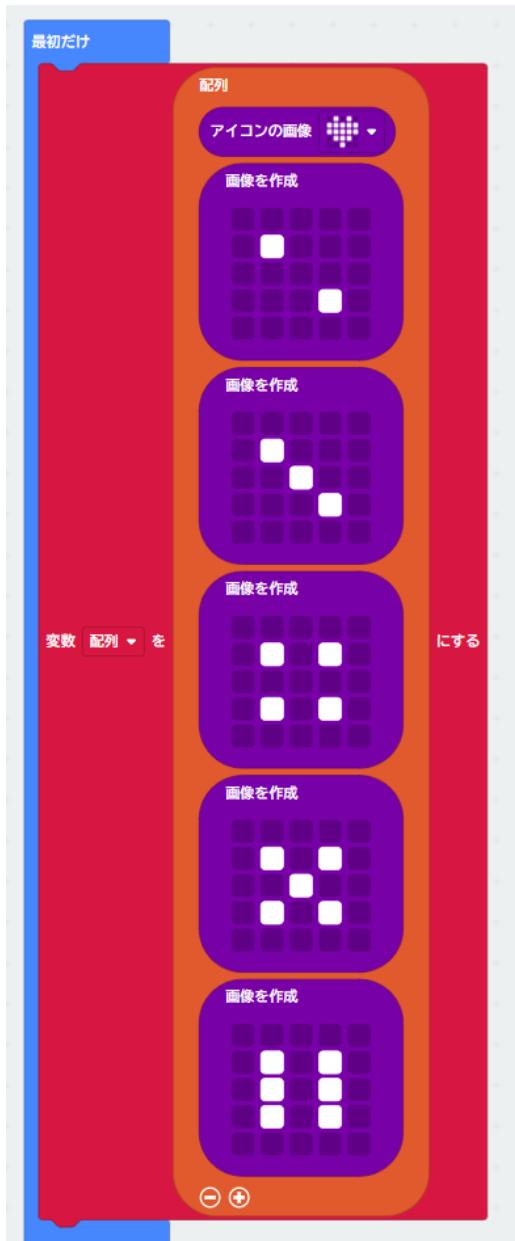
・画面が上になった時、メロディも鳴るようにしてみよう！

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

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

micro: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。

- ・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



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

★☆☆

「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

使うブロック

■ 基本

アイコンを表示

数を表示

■ 計算

0 から 10 までの乱数

■ 入力

ボタン A が押されたとき

ゆさぶられた とき

♠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

変数 JANKEN を 0 にする

JANKEN ▾

変数 JANKEN を 1 だけ増やす

②グー・チョキ・パーのアイコンをつくる



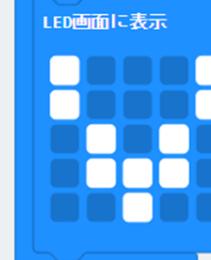
自分でデザインしてみよう！

値	アイコン
1	グー
2	チョキ
3	パー

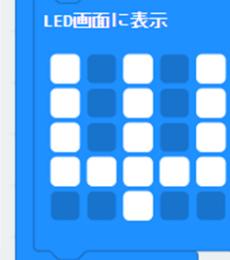
1 : グー



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

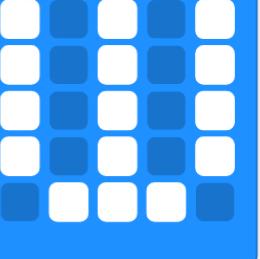
LED画面に表示



ずっと 3 ⇒ パー

もし JANKEN ▾ = ▾ 3 なら

LED画面に表示

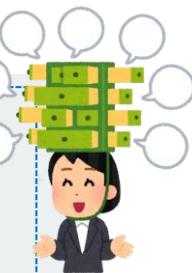


変数 = 「変わらる数」

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。

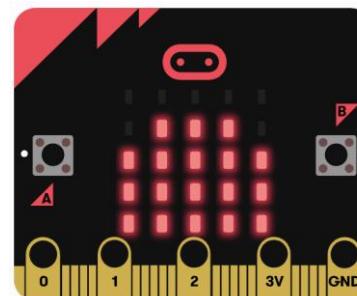
そこで、磁力を音や光に変えるプログラムを作り、

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！



磁力が
強い=バーが多い
弱い=バーが少ない

棒グラフで探す

★☆☆

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





⑤ 音の高低で探す！(v2用)

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

(+) 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 (μ T) 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！

<https://nextday-kids.com/wp/1112/>

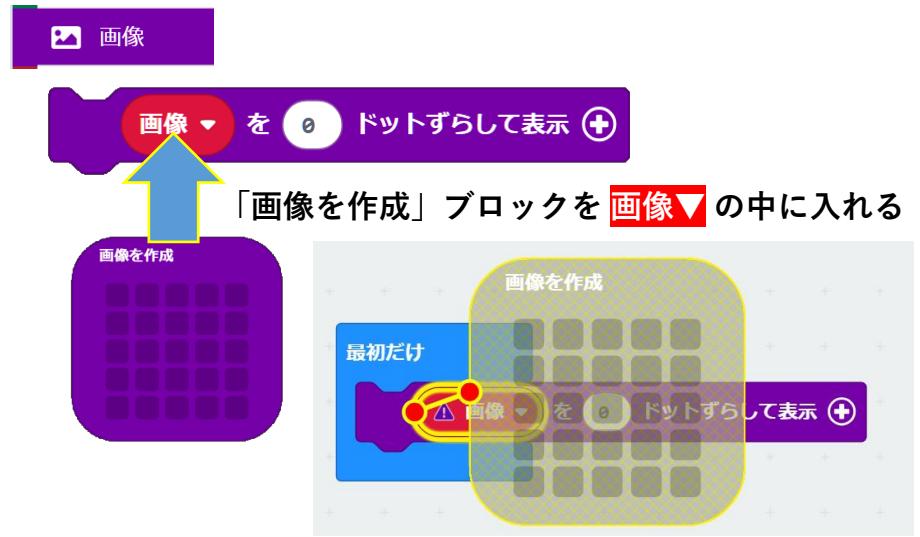
★★★

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

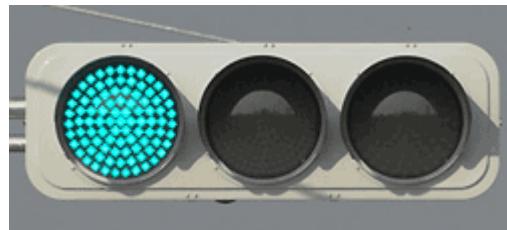
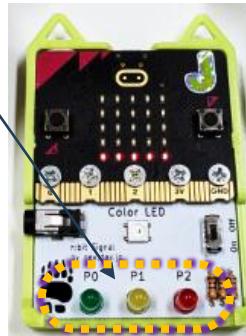
- 最初は、車用は緑「●」、歩行者用は「止まれ」を表示させます。
- ボタンAが押されたなら、まず、車用を黄「○」にします。
- 2秒後に、車用を赤「●」にします。
- さらに2秒後に、歩行者用を「すすめ」の表示にします。
- 5秒後に歩行者用の「すすめ」表示を6回点滅させます。
- 歩行者用を「止まれ」の表示にします。
- 1秒後に車用を緑「●」にします。

①歩行者用画像ブロック [止まれ] をつくる

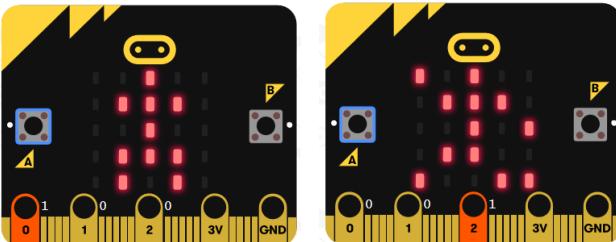


3色 LED

端子	色
P0	緑
P1	黄
P2	赤



車用信号機
歩行者用信号機



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〕ことが必要です。
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

P0 P1 P2 P3
P4 P5 P6 P7
P8 P9 P10 P11
P12 P13 P14 P15
P16 P19 P20

値 状態

0 : 消える
電気を〔止める〕

1 : 点く
電気を〔流す〕



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
デジタルで出力する 端子 P1 値 1
一時停止 (ミリ秒) 2000
デジタルで出力する 端子 P1 値 0
デジタルで出力する 端子 P2 値 1
一時停止 (ミリ秒) 2000

緑 (P0) が消える

黄 (P1) が点く

黄 (P1) が消える
赤 (P2) が点く

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

デジタルで出力する 端子 P0 値 1

赤 (P2) が消える

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



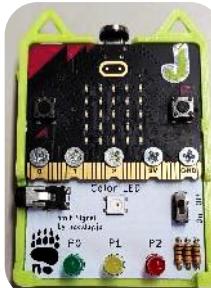
ライントレース
ロボットカー
を走らせよう！

n:bit II

8連RGB LED



※3.5mmオーディオケーブルで
micro:bitと接続

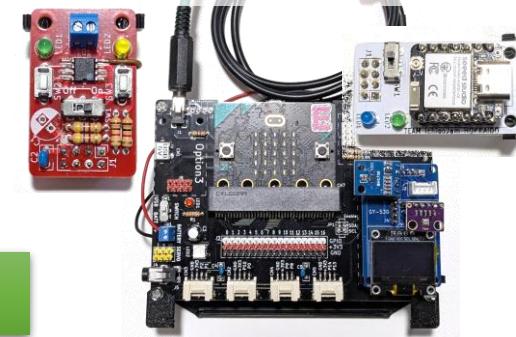


透明アクリル板やペットボトルキャップ
をキャンバスにしてイラストを描く。



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)

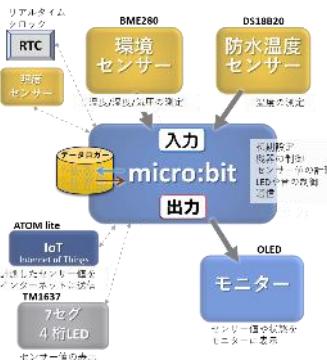


micro:bit用
ワークショップモジュール

nextday プログラミングきっづ

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