公開講座
マイコンプログラミングで課題解決

鳥羽商船高専
情報機械システム工学科

鳥羽商船高等専門学校について

池の浦駅から
歩いて
約10分！

三重県鳥羽市にある高等専門学校

高専

独立行政法人国立高等専門学校機構
鳥羽商船高等専門学校

高等専門学校とは？

高校1年生から5年間で

おもに**専門技術**(**プログラミング**

ロボット、電気、機械など)を

学ぶことができる学校

今日の流れ

１．課題解決とは？

２．micro:bitの使い方

　　プログラミングの練習「順接」「分岐」「反復」

３．温湿度センサーを使ってみよう

　（micro:bitで暑さ指数を表示）

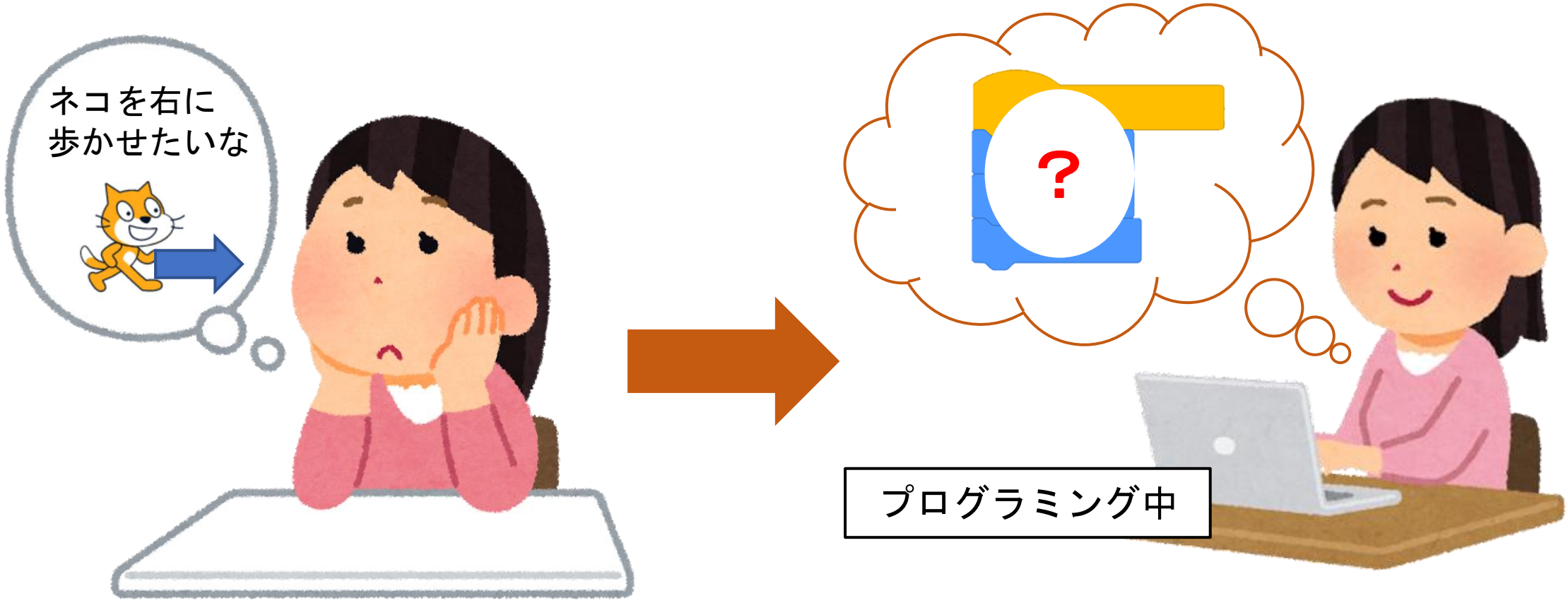
４．身近な問題をプログラミングで解決しよう

　（温度センサで扇風機を制御）

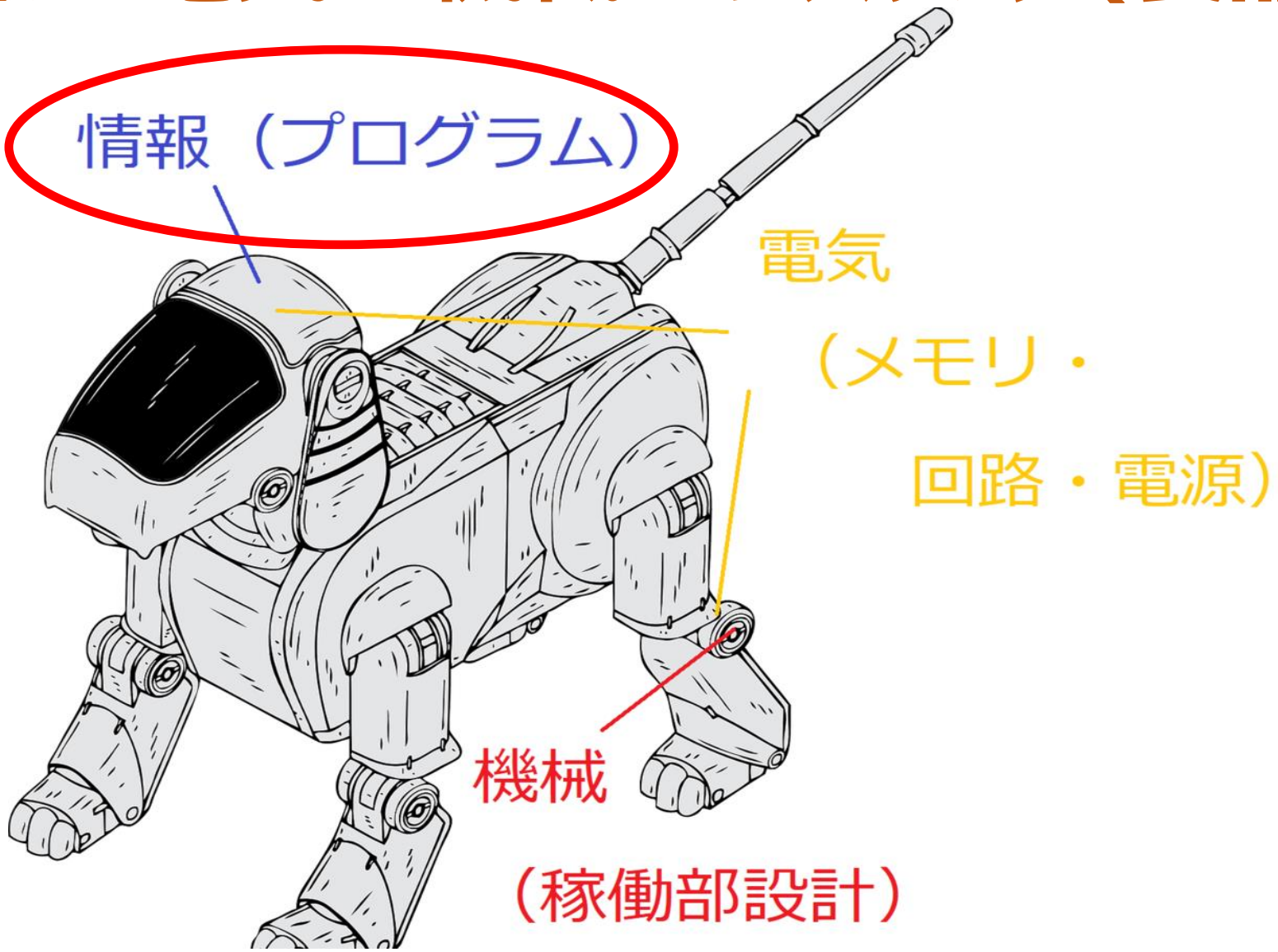
５．説明スライドの作成

プログラミングって何？

機械に「こういう風に動いてほしい」という指示を書くこと



情報＋電気＋機械＝システム（製品）



なぜプログラミングが必要なの？

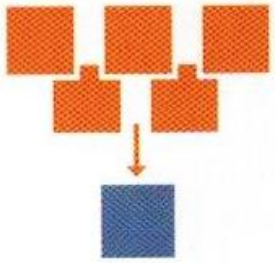
理由

将来どのような職業についても、
必ず求められる力だから。

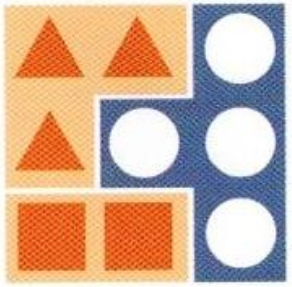
プログラミング的思考

コンピュータやプログラムの概念に
もとづいた“問題解決”型の思考

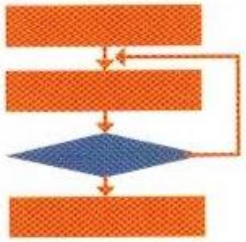
分解



抽象化



一般化



繰り返し

×

情報を上手に使う力

情報を収集・整理・分析して、
人に伝達していく能力

情報デザイン

パソコン・
タブレットの使い方

情報収集

情報モラル・
セキュリティ

出展：株式会社ベネッセコーポレーション

カレーの作り方はプログラミング？

課題　→　カレーを作る

始め

材料を用意する

カレールーを入れる

シチューのルーを入れる

材料を切る

お店でカレーを注文する

いためる

水を入れる

にる　にる

終了

カレーの作り方

・順番、やることがちがう　→完成しない、遅い、違う料理

・正しい順番　→　だれでもカレーが作れる

→　プログラミングのように「だれもが分かる言葉」＋
「正しい順序」で説明すると、みんなに分かりやすく伝わる。
説得力UP、コミュニケーション力UP

プログラミングを学ぶとそれが身に付く
→論理的（ろんりてき）思考・プログラミング的思考

作り方

①いためる

玉ねぎがしんなりするまで

中火

厚手のなべにサラダ油を熱し、一口大に切った具材をいためる。

②水を入れ、煮込む

水1400ml

弱火～中火

あくを取り、材料が柔らかくなるまで煮込む。(沸騰後約15分)

③ルウを入れる

火を止める

ルウを割り入れて溶かす。

④煮込む

とろみがつくまで

弱火

再び煮込む。(約10分)

はちみつを入れる場合 作り方②で入れ、沸騰後20分以上煮込む。

出展：ハウス食品グループHP

課題解決は、日常でも使用している

課題：家族は6人です。1人3個ずつリンゴをもらいました。
　　全部で何個もらいましたか？

・課題の分解　何を聞かれている？　→リンゴ全部の数
　　分かっていることは？　→6人家族
　　1人3個りんごをもらった
　　何算になる？単位は？　→掛け算、「個」
・組み合わせ　式を立てる　→３個×6人＝18個
　　答え：18（個）

・一般化　1人あたりの個数：□個、家族：△人
　　→　□×△＝（答え）（個）

応用

課題　→　カレーライスを作る

始め

カレールーを入れる

米を洗う

にる

材料を用意する

材料を切る

いためる

にる

水を入れる

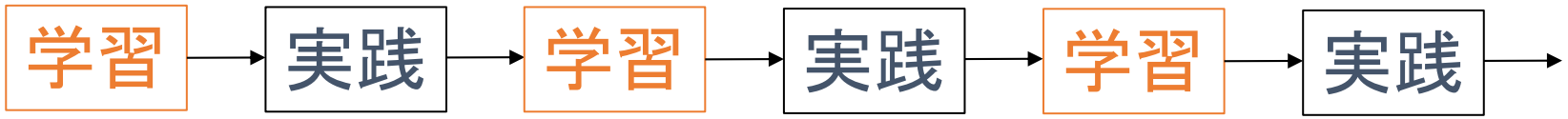
炊飯器のスイッチを押す

米と水を炊飯器に入れる

終了

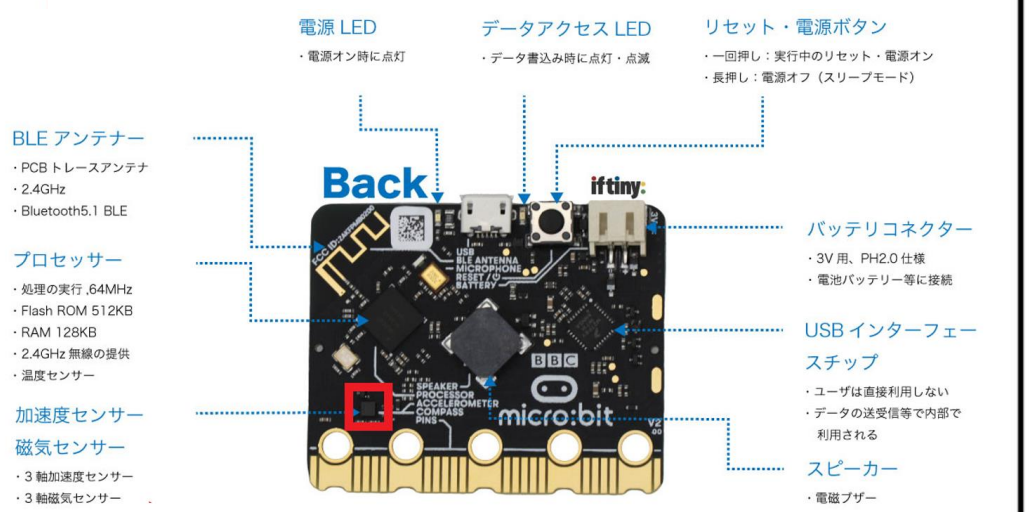
よりよく学ぶために

- 実際に手を動かしてブロックを並べたり、プログラムコードを入力する、一部を変更して実行する
- Webで公開されているプログラムコードを動かしてみる、一部を変更して実行する
- プログラミングコンテスト（U16プロコン）に参加してみる（モチベーションアップ、実力向上、他者のコードから学ぶ）



マイクロビット

- 子ども向けのプログラミング教材として作られたマイコン
- Scratchのように ブロックを繋げてプログラミングを行う
- イギリスでは 無償提供されている



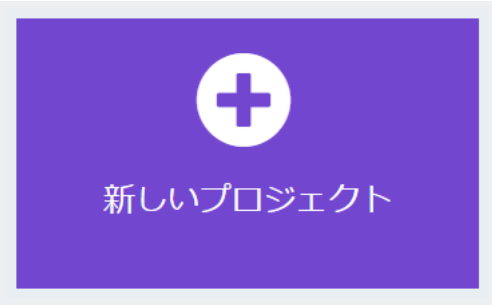
micro:bitの外観および搭載センサ

micro:bitを使ったプログラミング画面

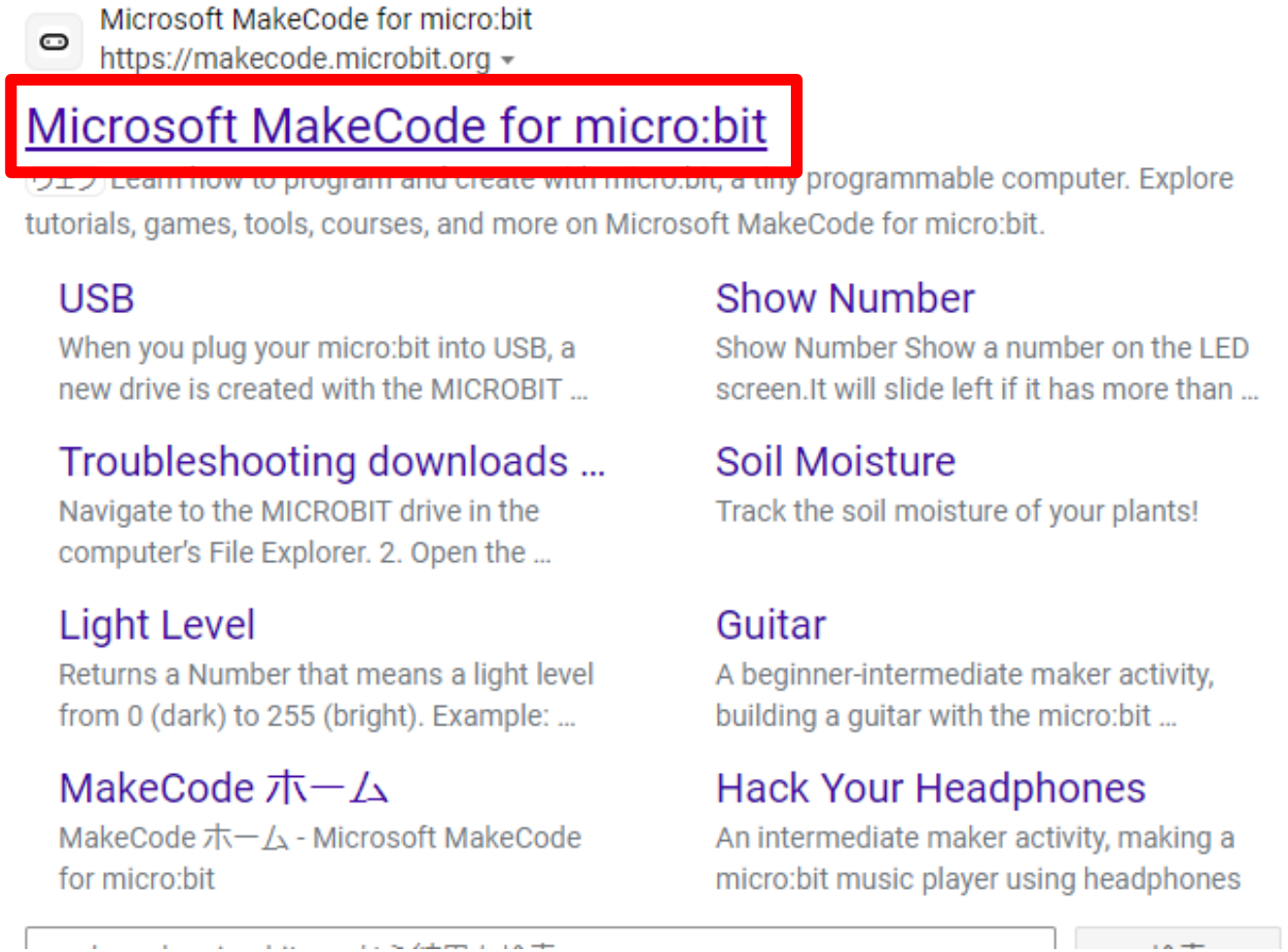
センサ（温度、加速度、磁気、照度、タッチ）出力（スピーカー、LED）通信（Bluetooth、シリアル）

micro:bitを開こう

1.「makecode」と検索！
2. 赤枠のページを開こう！
3. 新しいプロジェクトをクリック



4. プロジェクト名をつける

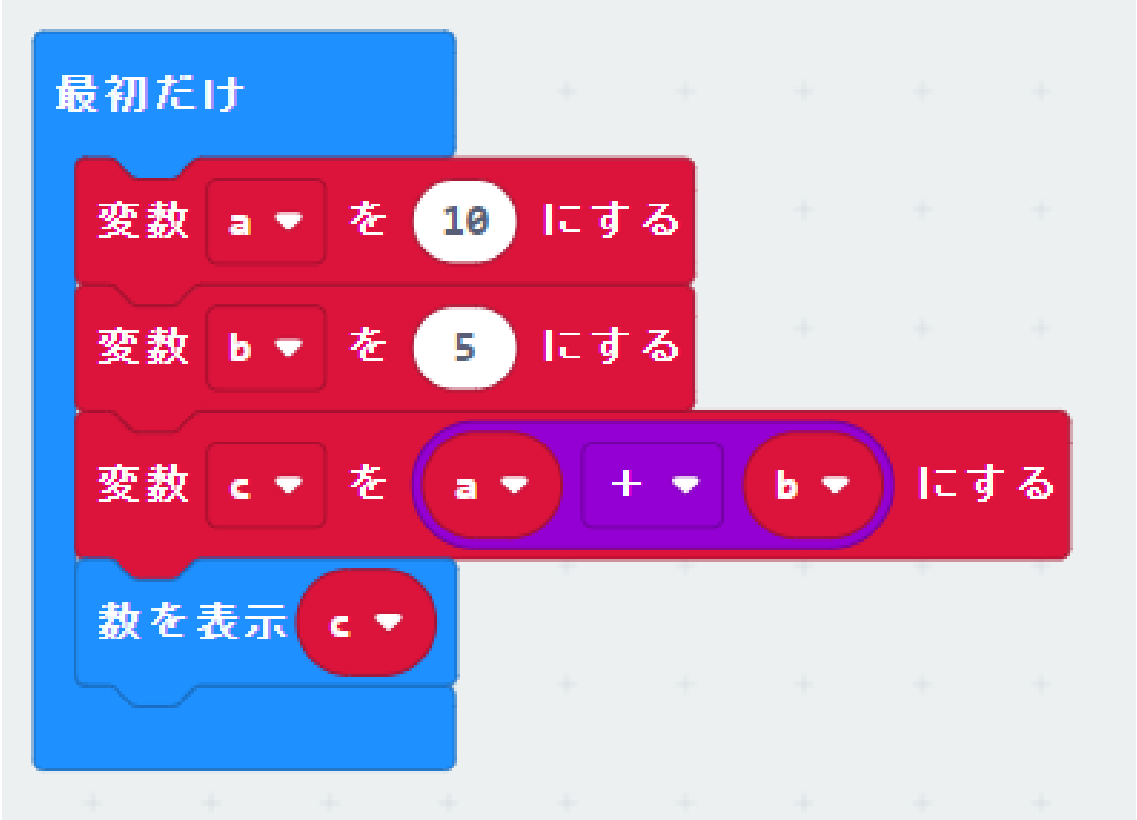


順接

足し算の結果をmicro:bit に表示しよう！

順接

2つの数字を足した結果を表示する



変数

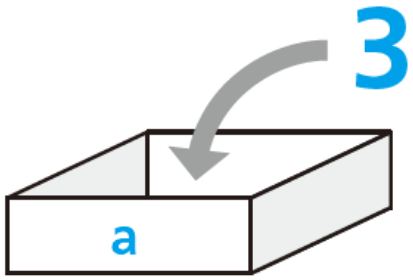
「**変数**」とは、値を入れておく入れ物

```
a = 3
```

←「変数aに3を**代入**する」

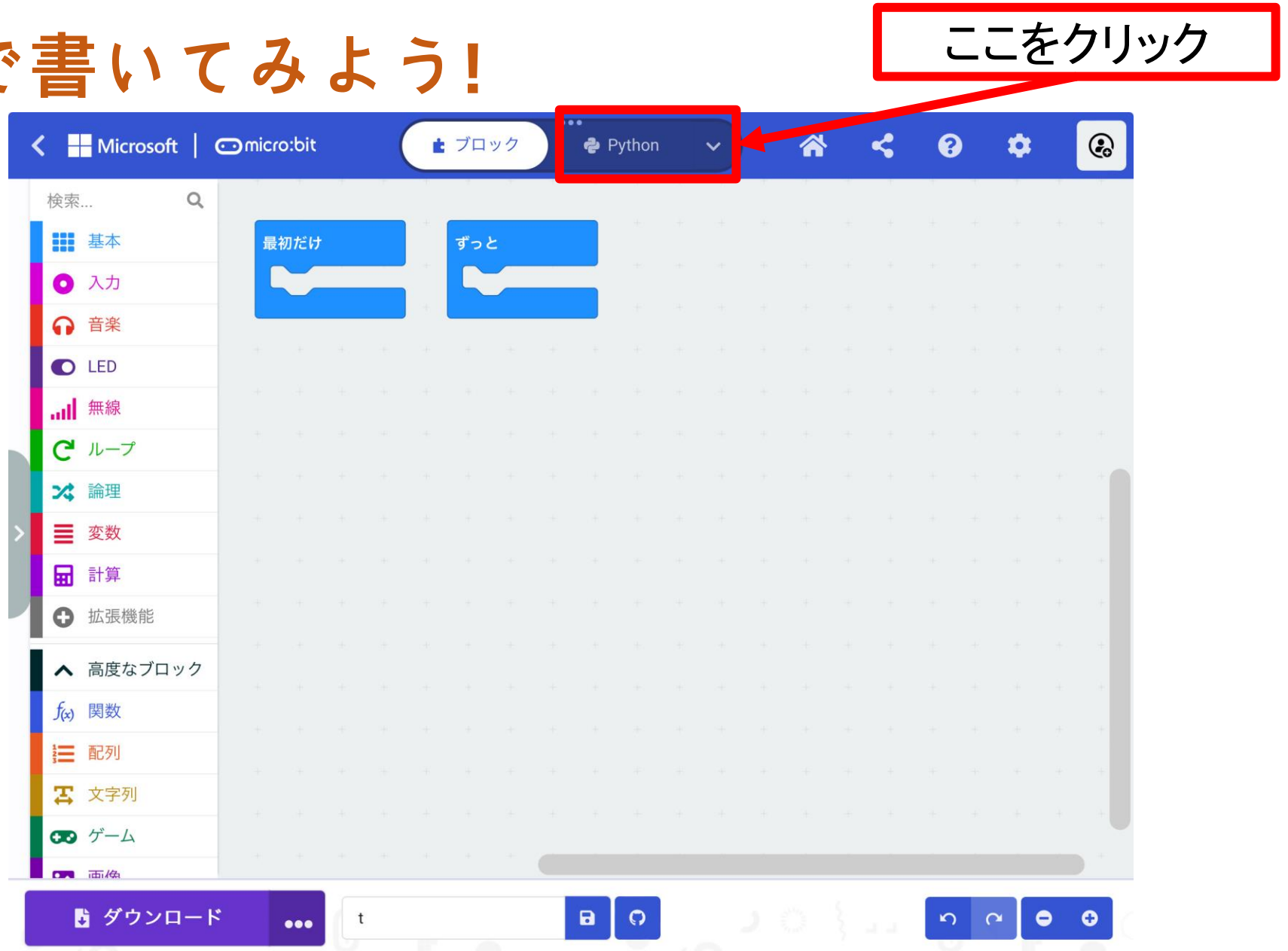
変数名

「代入」の一般的なイメージ



a という名前のついた箱に 3 を入れる

コードで書いてみよう!


ここをクリック
Microsoft
micro:bit
ブロック
Python
検索...
基本
入力
音楽
LED
無線
ループ
論理
変数
計算
拡張機能
高度なブロック
関数
配列
文字列
ゲーム
最初だけ
ずっと
ダウンロード
t

順接

2つの数字を足した結果を表示する

変数a,bに値を代入

変数cにa＋bを代入

変数cの値を
LED画面に表示する

```
1    a = 10
2
3    b = 5
4
5    c = a + b
6
7    basic.show_number(c)
8
```

演算子	演算の内容
+	加算（足し算）
-	減算（引き算）
*	乗算（掛け算）
/	除算（割り算）
//	商
%	剰余
**	べき乗
:=	代入

コード

分岐

数字の大きさを
判別しよう！

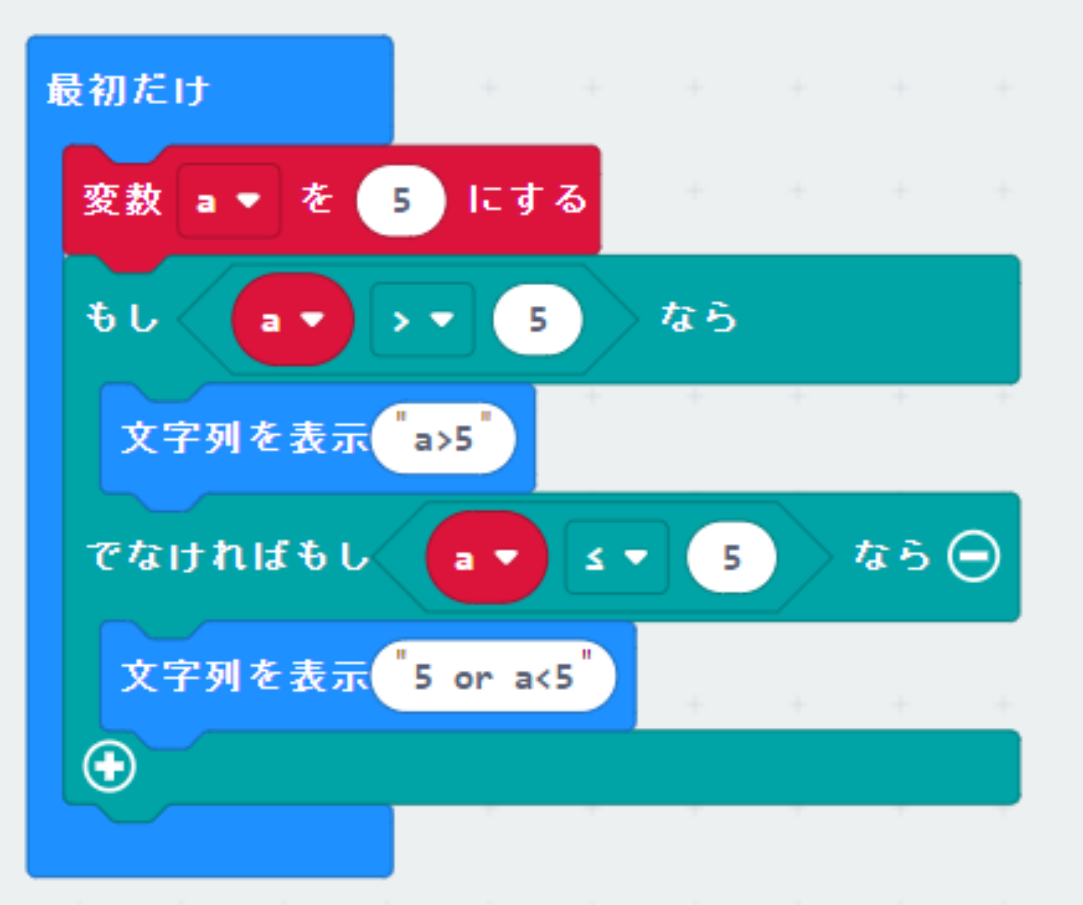
もう1ページmicro:bitを開こう

1.新しいページを開く「＋」をクリック
2.「micro:bit」と検索！
3. 先ほどと同じページを開こう！
4.新しいプロジェクトをクリック

2. プロジェクト名は
「扇風機」

分岐

数字の大きさを判定する



ブロック

分岐

比較演算子	意味
==	等しい
!=	等しくない
>	大きい
>=	以上
<	小さい
<=	以下

数字の大きさを判定する

変数aに値を代入

もしaが５より大きいなら“a>５”を表示する

そうでなければ“５ or a<５”を表示する

```
1    a = 5
2
3    if a > 5:
4        basic.show_string
5        ("a>5")
6
7    elif a <= 5:
8        basic.show_string
9        ("5 or a<5")
10
```

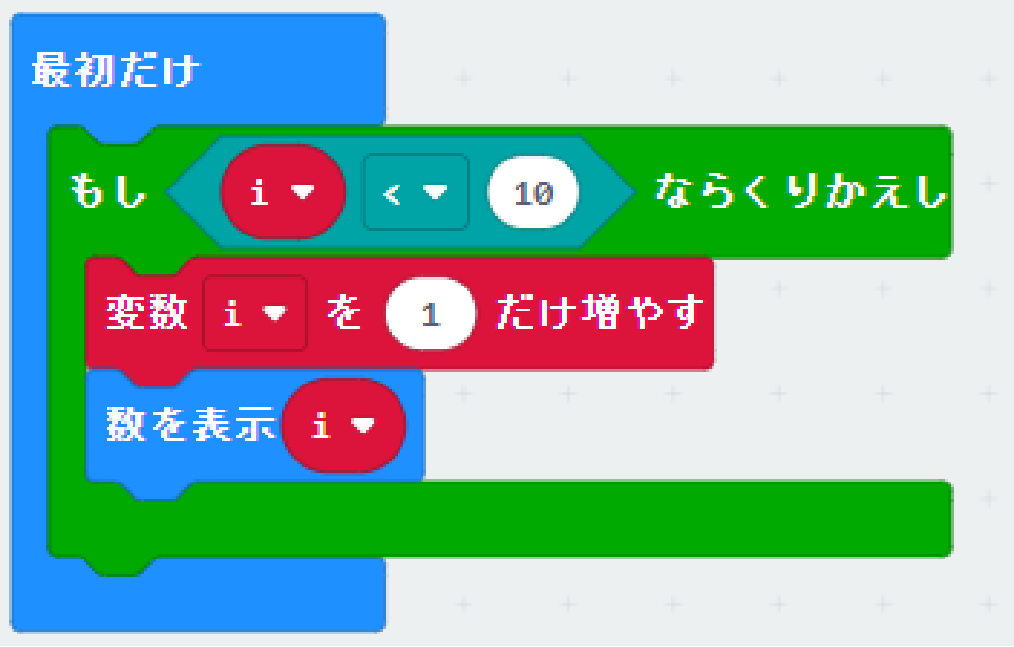
コード

反復

数字をカウントアップして
表示しよう！

反復

数字をカウントアップして表示する



ブロック

反復

数字をカウントアップして表示する

変数iを初期化する

iが10より小さいとき
iを+1してLED画面に
表示する

```
1    i = 0
2
3    while i < 10:
4        i += 1
5        basic.show_number(i)
6
```

コード

作成するもの

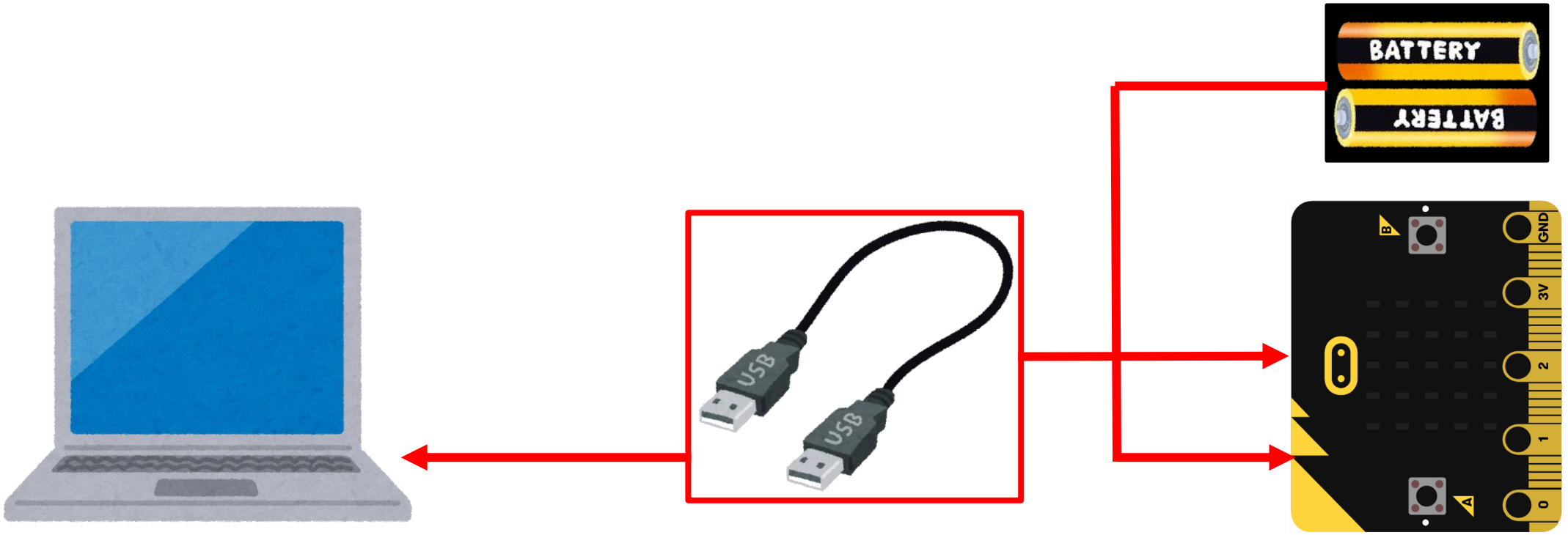
課題：
温湿度センサを用いて
暑さ指数を表示しよう！

配布物リスト

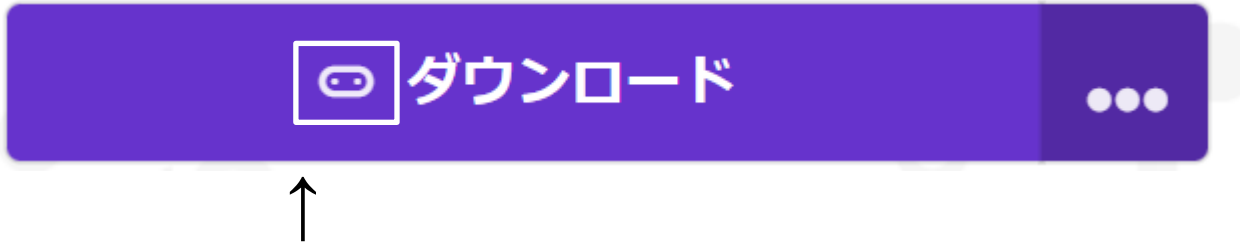
- micro:bit
- プロペラ
- モーター
- ワニロクリップ×３
- 結束バンド
- ドライバ
- Grove Shield
- リレーモジュールキット
- 温湿度センサー
- 電池入り電源ボックス×２
- 電池ボックススナップ
- USBケーブル

パソコンとmicro:bitを接続しよう

USBをパソコンとmicro:bitに接続しよう！
電池をmicro:bitに接続しよう！



実際にセンサなどを動かしてみよう!



↑

このマークになっていることを確認してダウンロードボタンをクリック！

ダウンロード出来たらパソコンからコードを抜き、きちんと動くか
確認しよう！

WBGTとは？

「暑さ指数（WBGT）」とは、気温だけではなく、湿度や風、日差しなどを考慮し、より体感に近い暑さ（体感温度）を表す指標

日常生活に関する指標

温度基準（WBGT）	注意すべき生活活動の目安	注意事項
危険（31℃以上）	すべての生活活動でおこる危険性	高齢者においては安静状態でも発生する危険性が大きい。外出はなるべく避け、涼しい室内に移動する。
厳重警戒（28～31℃※）		外出時は炎天下を避け、室内では室温の上昇に注意する。
警戒（25～28℃※）	中等度以上の生活活動でおこる危険性	運動や激しい作業をする際は定期的に充分に休息を取り入れる。
注意（25℃未満）	強い生活活動でおこる危険性	一般に危険性は少ないが激しい運動や重労働時には発生する危険性がある。

※日本生気象学会「日常生活における熱中症予防指針Ver.3」（2013）より

WBGT値早見表

早見表	相対湿度 %																
乾球温度℃	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100
40	29	30	31	32	33	34	35	35	36	37	38	39	40	41	42	43	44
39	28	29	30	31	32	33	34	35	35	36	37	38	39	40	41	42	43
38	28	28	29	30	31	32	33	34	35	35	36	37	38	39	40	41	42
37	27	28	29	29	30	31	32	33	35	35	35	36	37	38	39	40	41
36	26	27	28	29	29	30	31	32	33	34	34	35	36	37	38	39	39
35	25	26	27	28	29	29	30	31	32	33	33	34	35	36	37	38	38
34	25	25	26	27	28	29	29	30	31	32	33	33	34	35	36	37	37
33	24	25	25	26	27	28	28	29	30	31	32	32	33	34	35	35	36
32	23	24	25	25	26	27	28	28	29	30	31	31	32	33	34	34	35
31	22	23	24	24	25	26	27	27	28	29	30	30	31	32	33	33	34
30	21	22	23	24	24	25	26	27	27	28	29	29	30	31	32	32	33
29	21	21	22	23	24	24	25	26	26	27	28	29	29	30	31	31	32
28	20	21	21	22	23	23	24	25	25	26	27	28	28	29	30	30	31
27	19	20	21	21	22	23	23	24	25	25	26	27	27	28	29	29	30
26	18	19	20	20	21	22	22	23	24	24	25	26	26	27	28	28	29
25	18	18	19	20	20	21	22	22	23	23	24	25	25	26	27	27	28
24	17	18	18	19	19	20	21	21	22	22	23	24	24	25	26	26	27
23	16	17	17	18	19	19	20	20	21	22	22	23	23	24	25	25	26
22	15	16	17	17	18	18	19	19	20	21	21	22	22	23	24	24	25
21	15	15	16	16	17	17	18	19	19	20	20	21	21	22	23	23	24

日本生気象学会「日常生活における熱中症予防指針」Ver.1 2008.4

WBGT計算式（屋内）

$$WBGT = 0.735 \times 温度 + 0.0374 \times 湿度 + 0.00292 \times 温度 \times 湿度 - 4.05828$$

WBGTが31度以上なら危険！！
警告音を鳴らそう！

扇風機を動かす仕組み

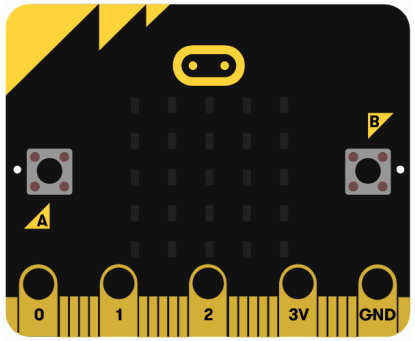
入力装置

温度、湿度の
値を読み込む

入力

温湿度センサ

計算・処理



micro:bit

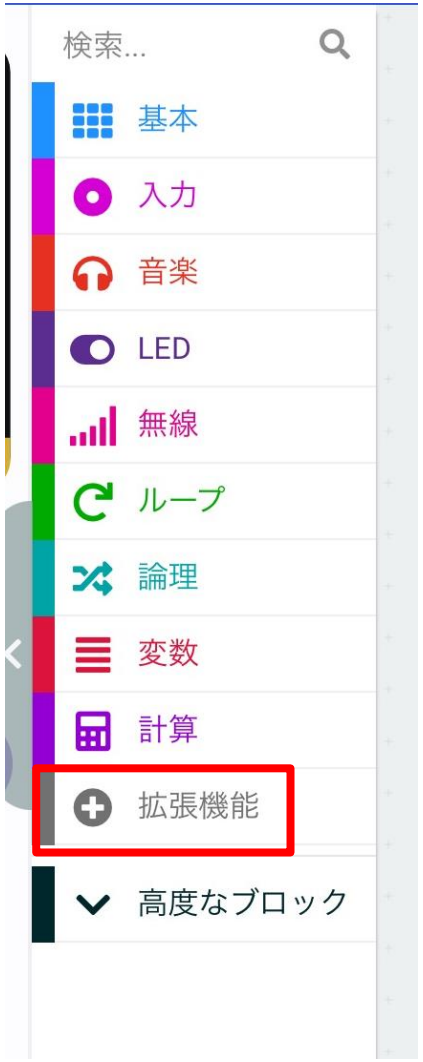
出力

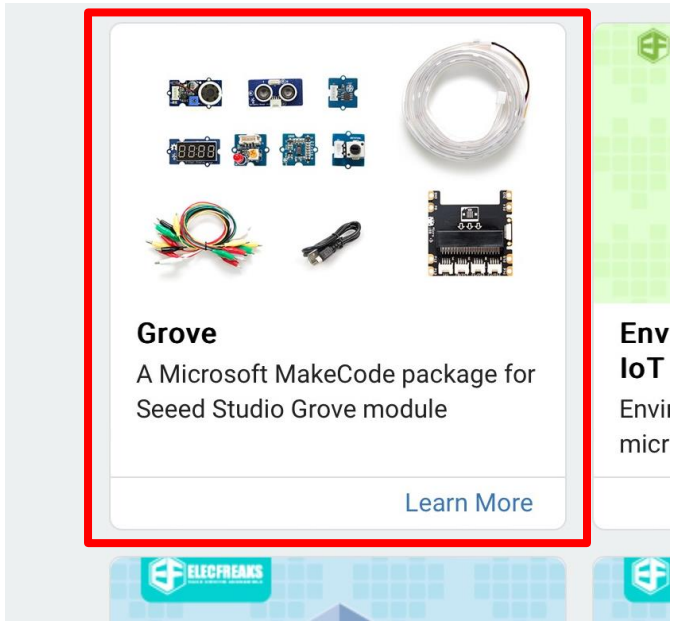
出力装置

LEDと音で警告

LED＋スピーカー

準備


検索...
基本
入力
音楽
LED
無線
ループ
論理
変数
計算
拡張機能
高度なブロック


Grove
A Microsoft MakeCode package for Seeed Studio Grove module
Learn More
ELECFREAKS


検索...
基本
入力
音楽
LED
無線
ループ
論理
変数
計算
Grove
拡張機能
高度なブロック

センサ値を読み取ろう

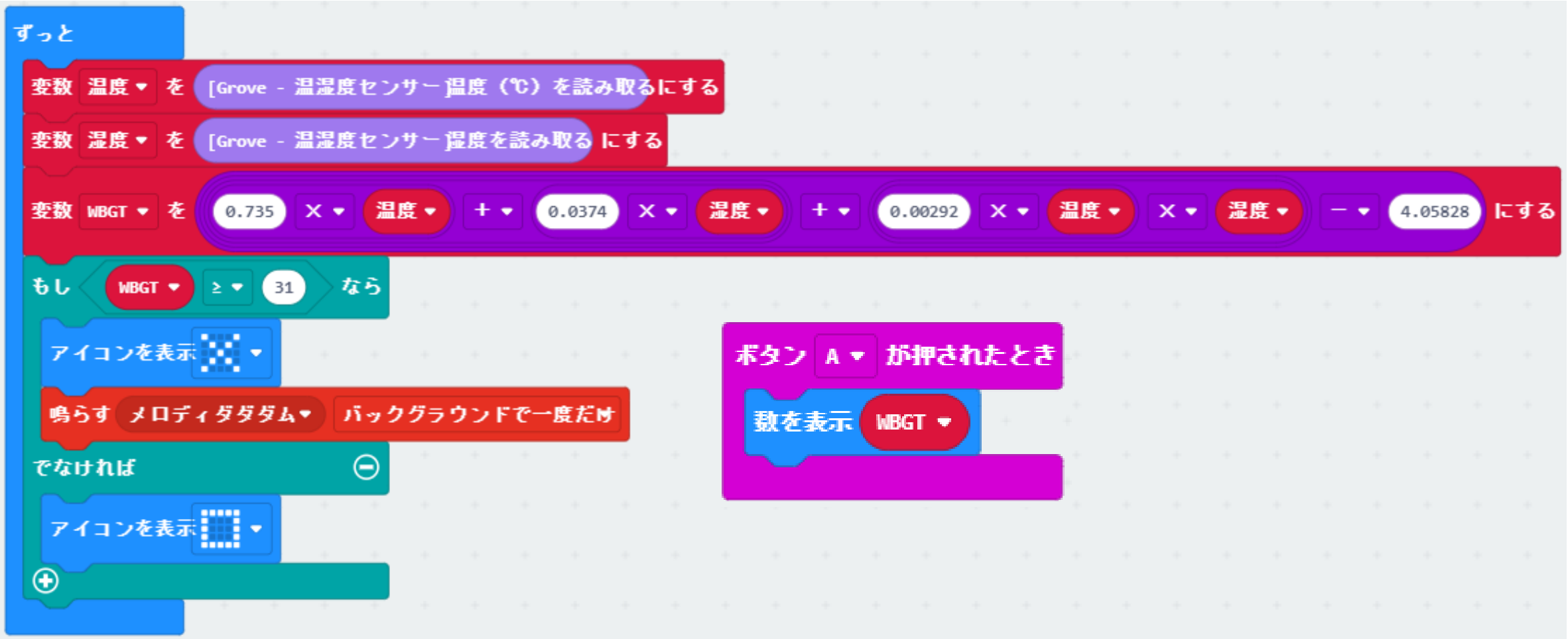

ずっと
シリアル通信 名前と数値を書き出す "temp" = [Grove – 温湿度センサー] 温度（°C）を読み取る
シリアル通信 名前と数値を書き出す "humi" = [Grove – 温湿度センサー] 湿度を読み取る
一時停止（ミリ秒） 100
temp: 25.74
25.71
50.77
humi: 50.44
50.35
temp:25.7284164428711
humi:50.4965782165527
temp:25.7064819335938
humi:50.443172454834
temp:25.7368087768555

WBGTを表示しよう！

温度と湿度を取得してシリアル通信で出力

```
 9    def on_forever():
10        serial.write_value("temp", grove.aht20_read_temperature_c())
11        serial.write_value("humi", grove.aht20_read_humidity())
12        basic.pause(100)
13    basic.forever(on_forever)
14
```

WBGTを表示しよう！


ずっと
変数 温度 を [Grove - 温湿度センサー]温度（℃）を読み取るにする
変数 湿度 を [Grove - 温湿度センサー]湿度を読み取る にする
変数 WBGT を 0.735 × 温度 + 0.0374 × 湿度 + 0.00292 × 温度 × 湿度 − 4.05828 にする
もし WBGT ≥ 31 なら
アイコンを表示
鳴らす メロディ ダダダム バックグラウンドで一度だけ
でなければ
アイコンを表示
ボタン A が押されたとき
数を表示 WBGT

WBGTを表示しよう！

変数を初期化する

Aボタンが押されたときWBGTを表示

温度と湿度を取得してWBGTを計算

WBGTが31度以上のとき×を表示して音楽を鳴らす

WBGTが31度未満のとき〇を表示する

```
1  WBGT = 0
2  温度 = 0
3  湿度 = 0
4
5  def on_button_pressed_a():
6      basic.show_number(WBGT)
7  input.on_button_pressed(Button.A, on_button_pressed_a)
8
9  def on_forever():
10     global 温度, 湿度, WBGT
11     温度 = grove.aht20_read_temperature_c()
12     湿度 = grove.aht20_read_humidity()
13     WBGT = 0.735 * 温度 + 0.0374 * 湿度 + 0.00292 * 温度 * 湿度 - 4.05828
14     if WBGT >= 31:
15         basic.show_icon(IconNames.NO)
16         music._play_default_background(music.built_in_playable_melody(Melod
17             music.PlaybackMode.IN_BACKGROUND)
18     else:
19         basic.show_icon(IconNames.SQUARE)
20 basic.forever(on_forever)
21
```

WBGTを表示しよう！

温度と湿度を取得してWBGTを計算

WBGTが31度以上のとき×を表示して音楽を鳴らす

WBGTが31度未満のとき〇を表示する

```
15  def on_forever2():
16      global 温度, 湿度, WBGT
17      温度 = grove.aht20_read_temperature_c()
18      湿度 = grove.aht20_read_humidity()
19      WBGT = 0.735 * 温度 + 0.0374 * 湿度 + 0.00292 * 温度 * 湿度 - 4.05828
20
21      if WBGT >= 31:
22          basic.show_icon(IconNames.NO)
23          music._play_default_background(music.built_in_playable_melody(Mel
24              music.PlaybackMode.IN_BACKGROUND)
25
26      else:
27          basic.show_icon(IconNames.SQUARE)
28  basic.forever(on_forever2)
```

作成するもの

温度センサを用いて
扇風機を制御しよう！

扇風機を動かす仕組み

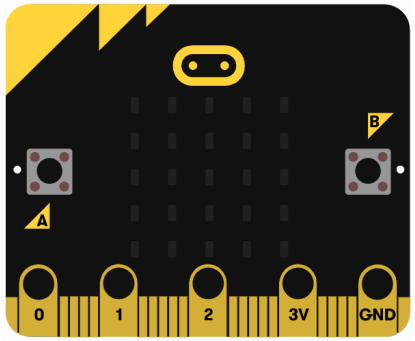
入力装置

計算・処理

出力装置

温度の
値を読み込む

プロペラのモータを
動かす指示を送る



温度センサ
（内臓）

入力

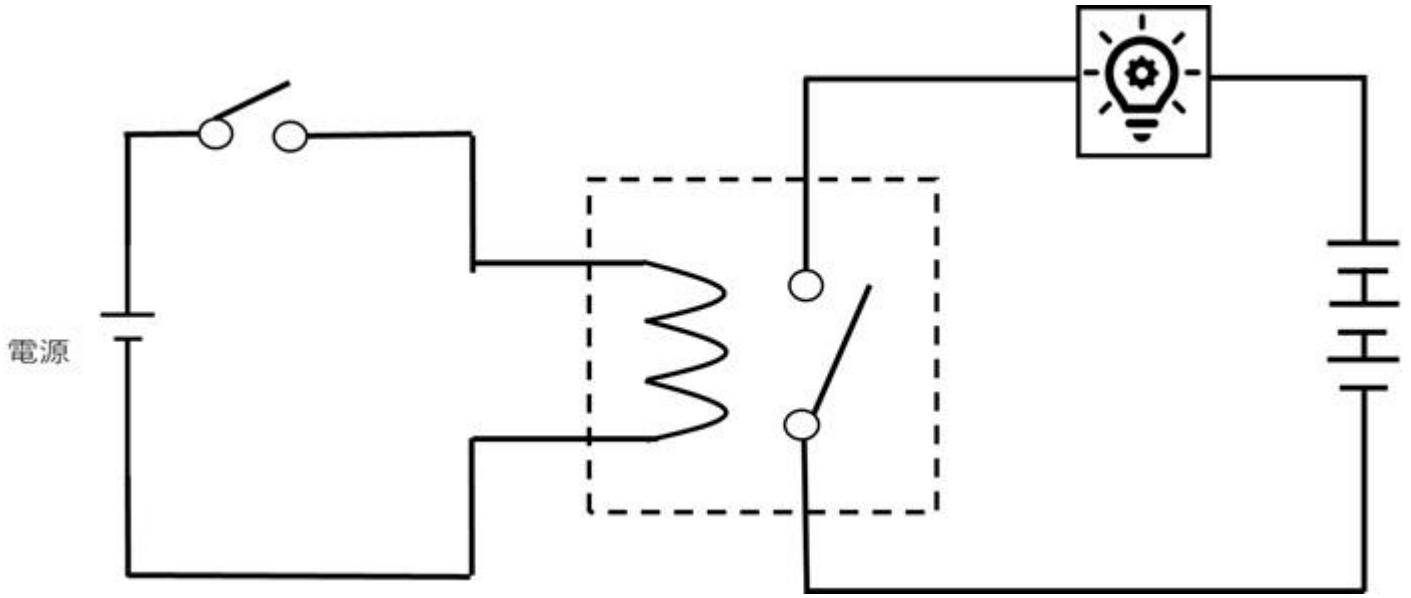
出力

micro:bit

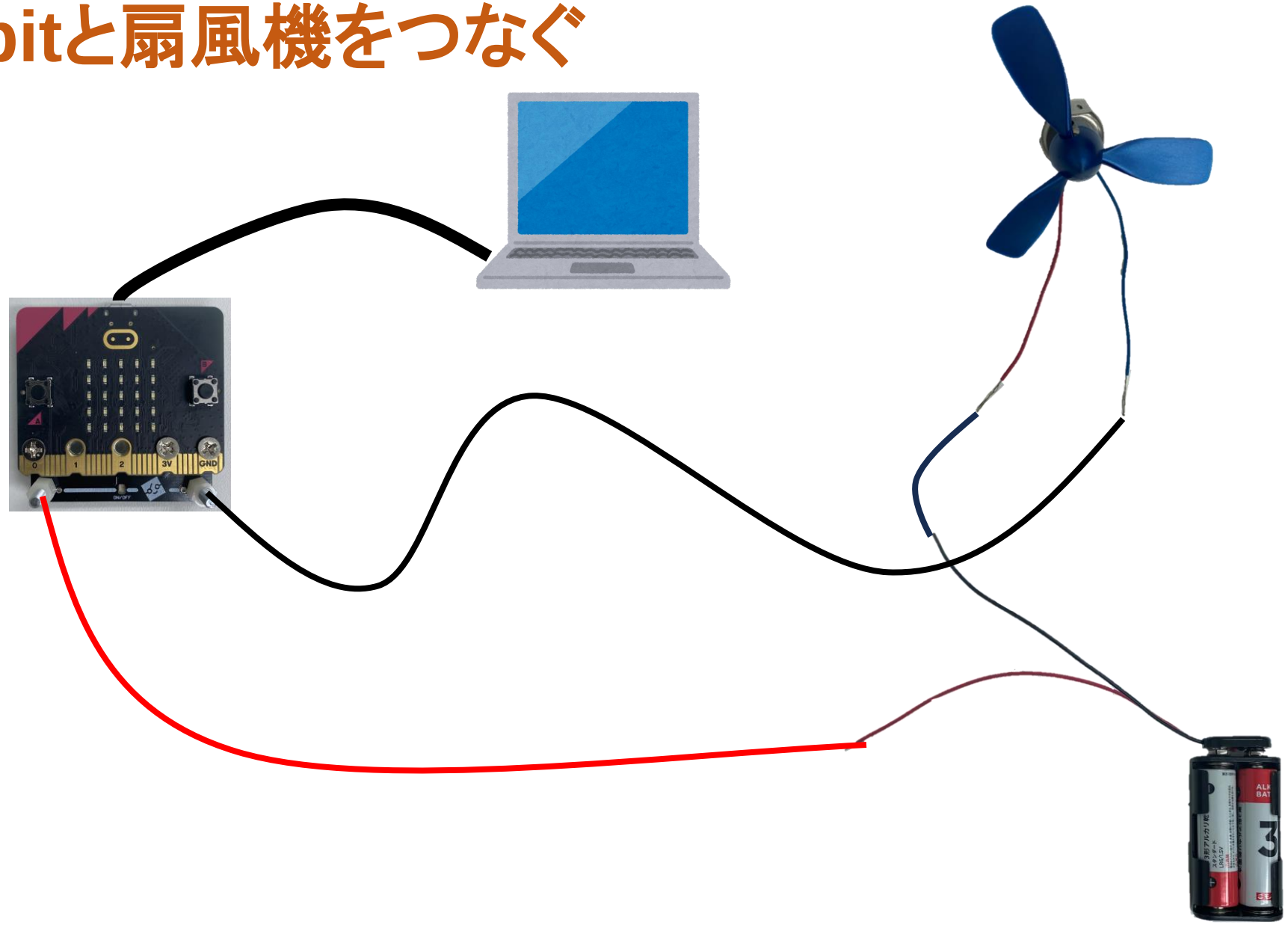
リレー＋モータ

リレー

リレーとは外部から電気信号を受け取り、
電気回路のオン/オフや切り替えを行う部品



micro:bitと扇風機をつなぐ



温度センサを用いて扇風機制御

- 変数「温度」を温度（℃）にする
- 温度が25度以上になったら

  →P0ピンを1にする（デジタルで出力する）

  →〇を表示
- 温度が25度以上でなければ

  →P0ピンを0にする（デジタルで出力する）

  →×を表示
- Aボタンが押されたら

  →温度（℃）を表示

温度センサを用いて扇風機制御

答え→

温度センサを用いて扇風機制御

温度を初期化

ボタンAが押されたとき温度を表示

温度を更新

25度以上ならP0ピンをオン、〇を表示

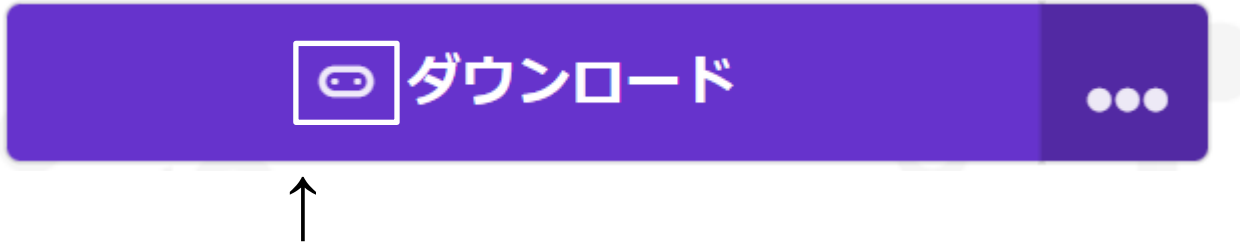
25度以上でないならP0ピンをオフ、×を表示

```
温度 = 0

def on_button_pressed_a():
    basic.show_number(input.temperature())
input.on_button_pressed(Button.A, on_button_pressed_a)

def on_forever():
    global 温度
    温度 = input.temperature()
    if 温度 > 25:
        pins.digital_write_pin(DigitalPin.P0, 1)
        basic.show_icon(IconNames.SQUARE)
    else:
        pins.digital_write_pin(DigitalPin.P0, 0)
        basic.show_icon(IconNames.NO)
basic.forever(on_forever)
```

実際に動かしてみよう!



↑

このマークになっていることを確認してダウンロードボタンをクリック！

ダウンロード出来たらパソコンからコードを抜き、扇風機がきちんと動くか確認しよう！

センサなどの購入先

秋月電子通商：
https://akizukidenshi.com/catalog/default.aspx
・micro:bit、Grove用センサなど
・電子部品など

スイッチサイエンス
https://www.switch-science.com/
・micro:bit関連製品

Amazon：
・「Grove　センサ」などで検索

参考となるサイト

MSR合同会社 「【連載】micro:bitを使ってみる」
https://msr-r.net/

Seeed K.K. エンジニアブログ
https://lab.seeed.co.jp/entry/2020/05/22/120000

その他にも「micro:bit」「Groveシールド」「センサ」
などのキーワードで検索すると参考になる

最後に
アンケートにご協力お願いします！