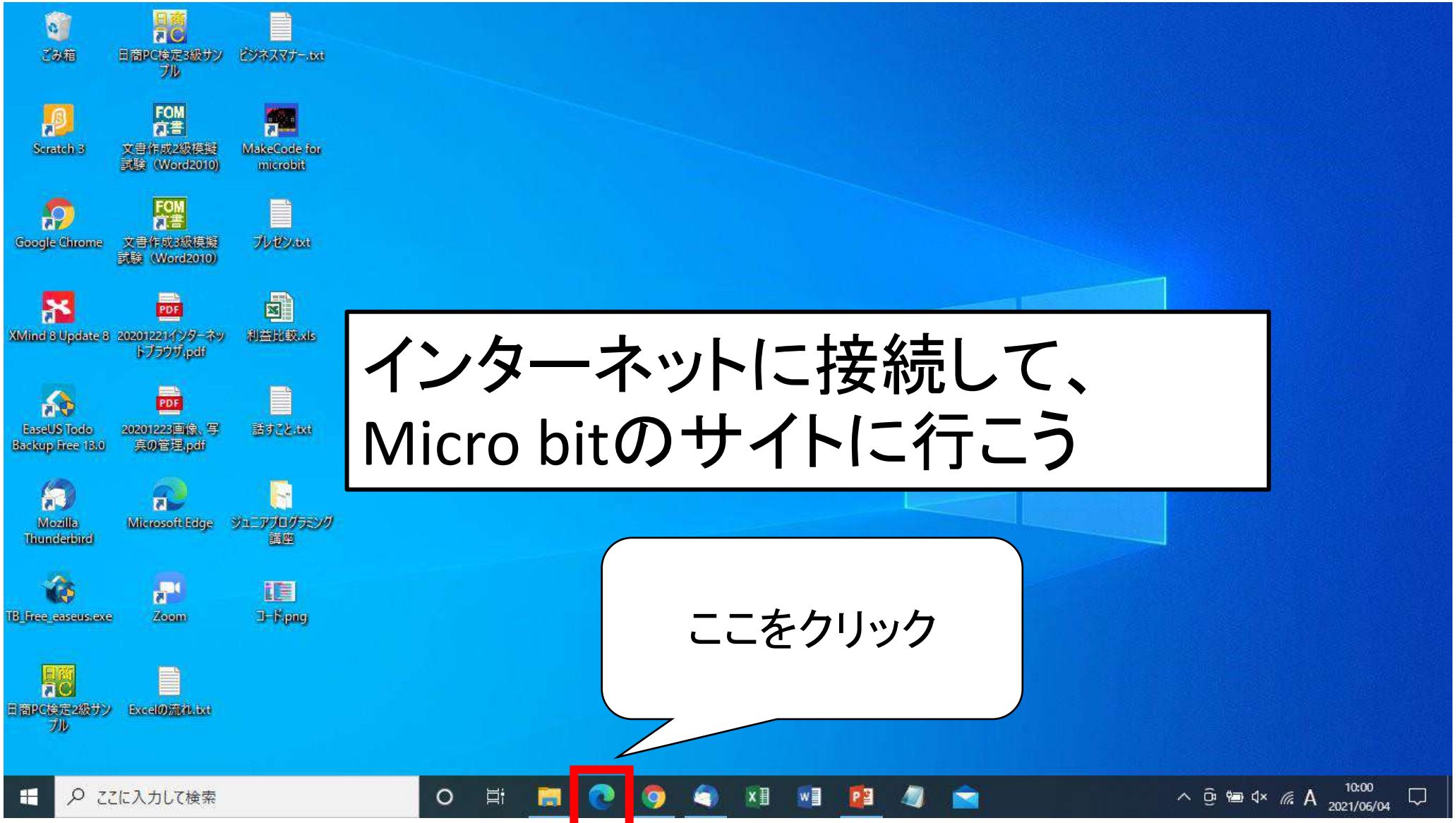
プログラミングをしてみよう

Micro bitにおけるプログラミング

Micro bit では、プログラミングをやったことがない人でも
ブロックを組み合わせることで簡単にプログラムを作成して
学ぶことができます。

プログラミング言語はむずかしい言葉ですが、実際に操作してみることで楽しみながらプログラミングの基本を学習していきましょう

Micro bitに接続する



Micro bitに接続する

ブラウザが開いたら、上の白いバーに
「Micro bit」と入力しよう

Micro bitに接続する

検索結果に「Microsoft MakeCode for micro:bit」というサイトがあるのでクリックしよう

Micro bitに接続する

このサイト上でプログラミングが出来る
「新しいプロジェクト」をクリックしてみよう

基本操作を学ぼう

１つのプログラムを「プロジェクト」と呼ぶ
「テスト」と名前を付けて作成ボタンを押そう

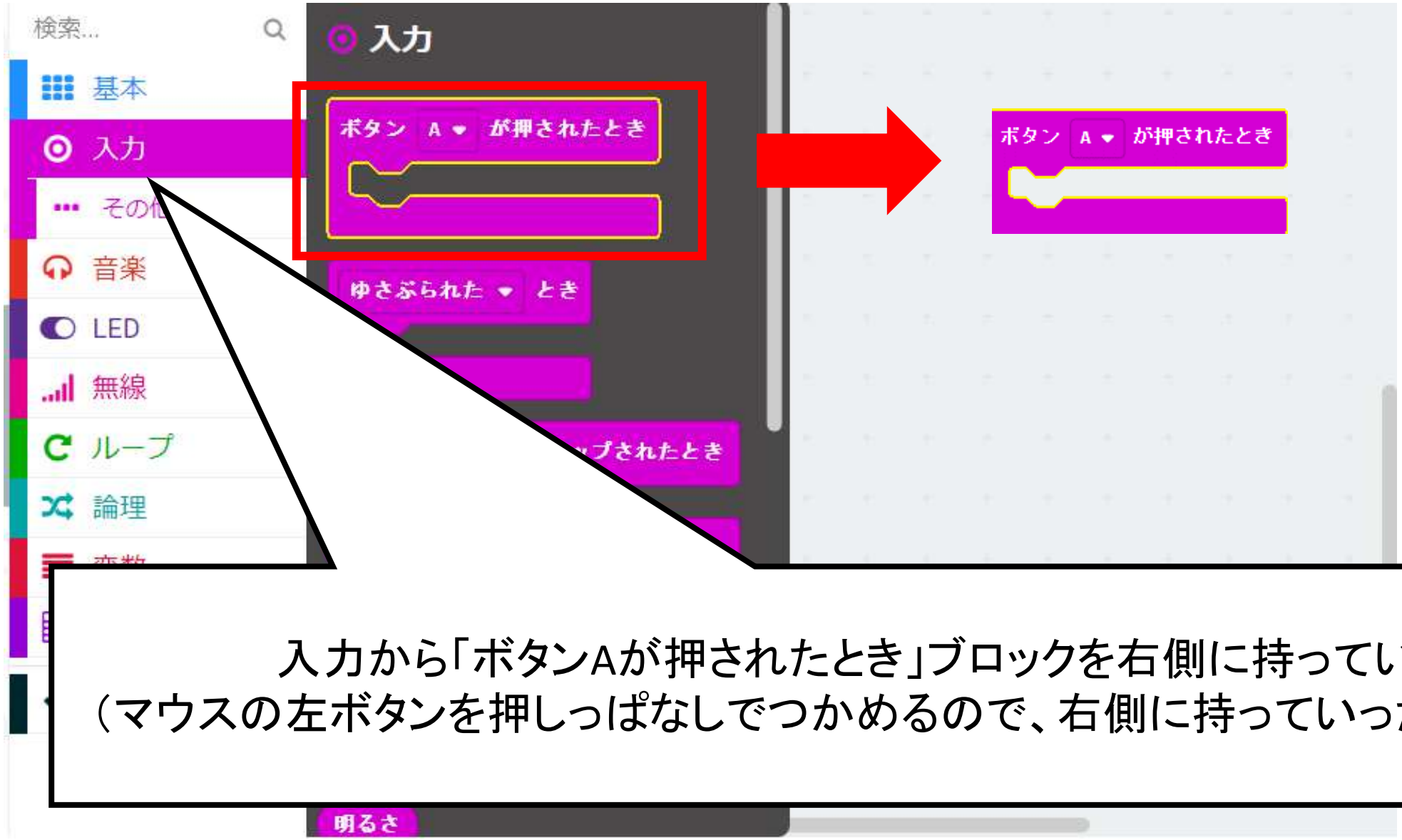
基本操作を学ぼう

実際にプログラムがどう動くか見れる

この中からプログラムブロックを取り出して

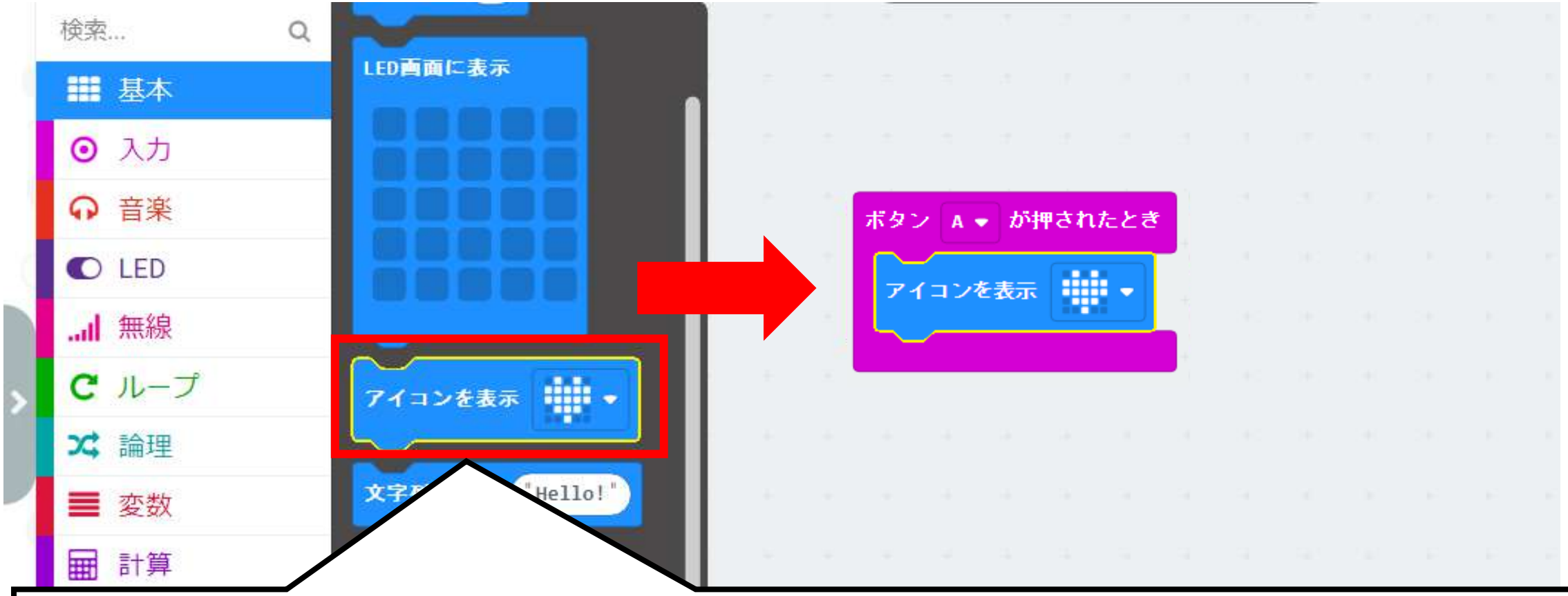
このエリアにブロックを配置
⇒組み合わせでプログラムを作成する

基本操作を学ぼう



入力から「ボタンAが押されたとき」ブロックを右側に持っていこう。
（マウスの左ボタンを押しっぱなしでつかめるので、右側に持っていったら指を放そう）

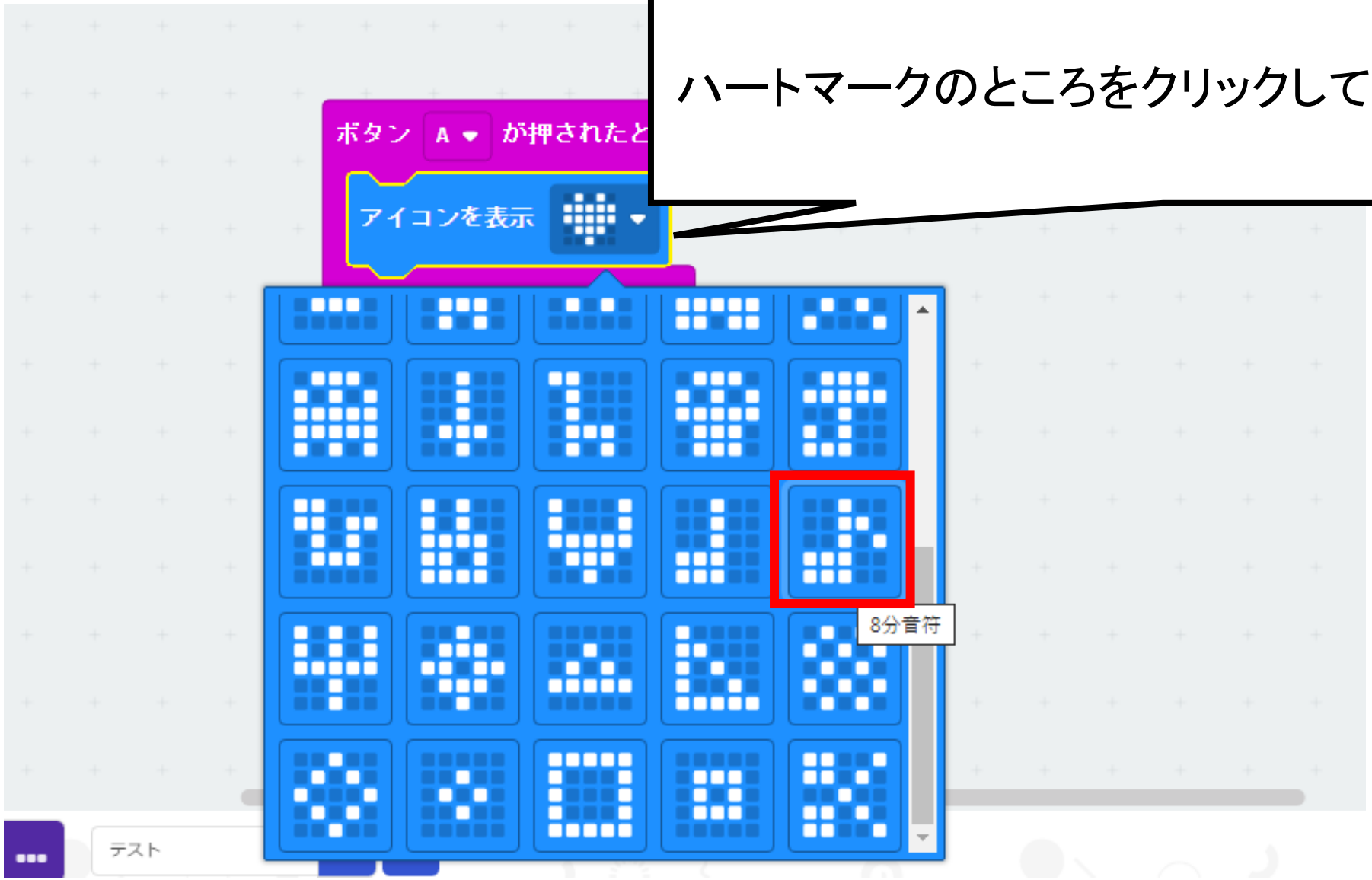
基本操作を学ぼう



基本から「アイコンを表示」ブロックを右側に持っていこう。
「ボタンAが押されたとき」ブロックと組み合わせよう。

基本操作を学ぼう

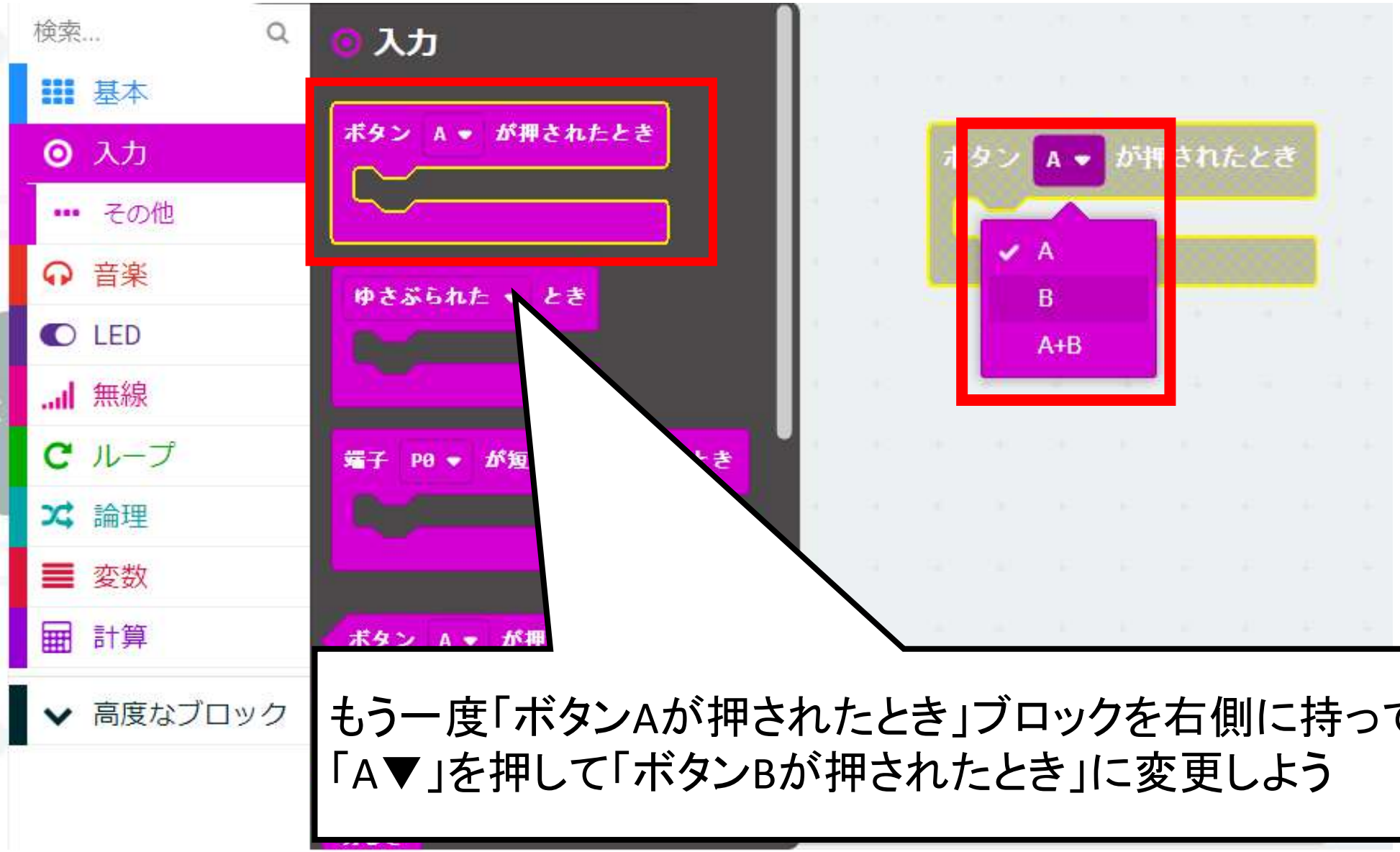
ハートマークのところをクリックして音符マークに変更しよう



基本操作を学ぼう

左側にあるMicro bitのAボタンをクリックしてみよう
これで、電源を入れてAボタンを押したらLEDが光る仕組みが出来たので、
次はBボタンを押して表示を消す仕組みを作ってみよう

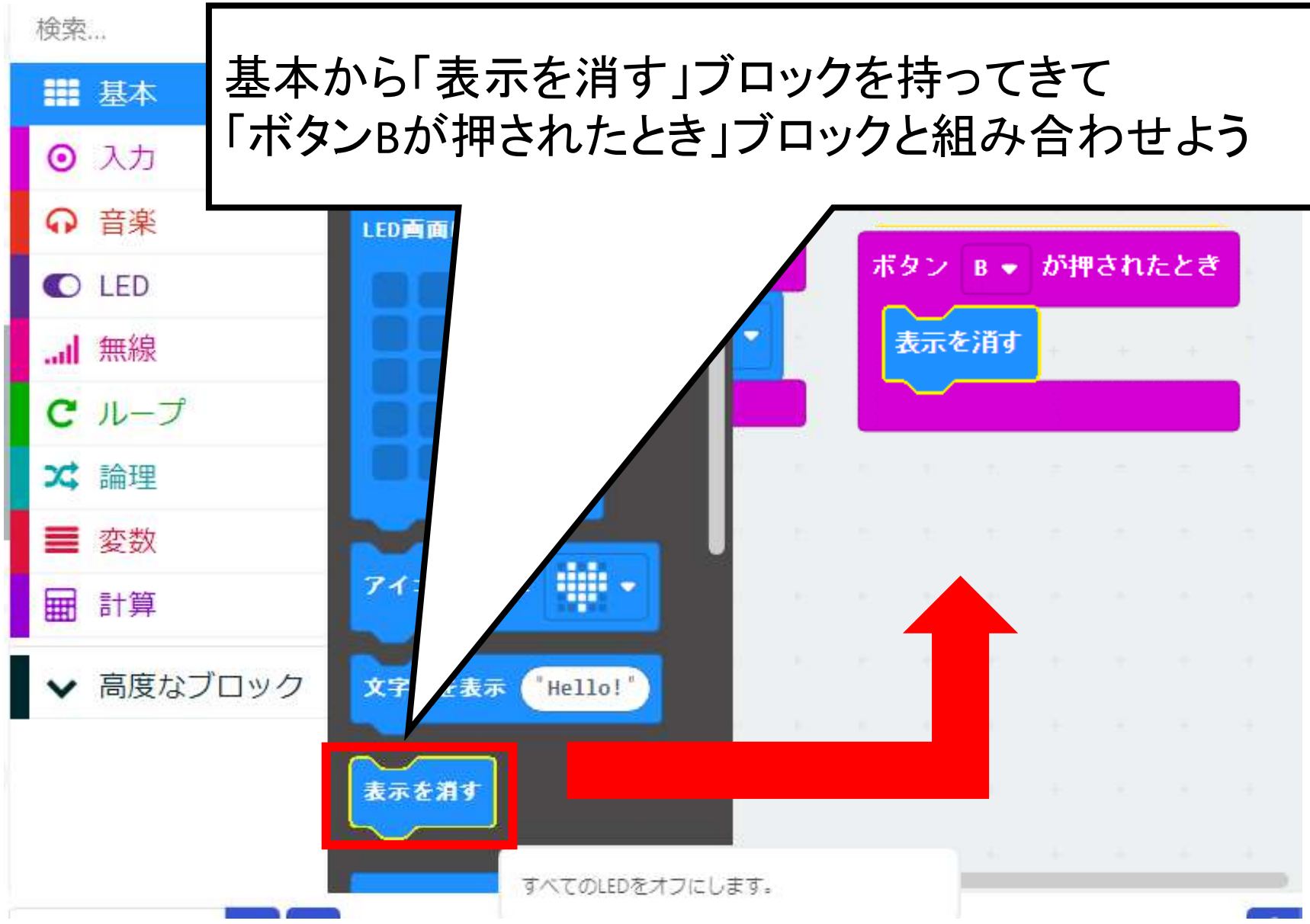
基本操作を学ぼう



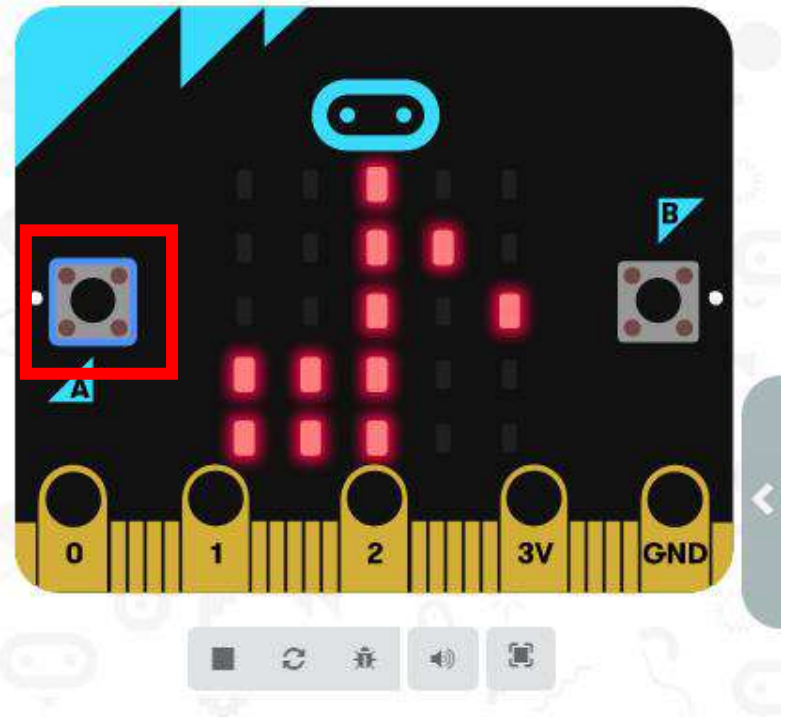
もう一度「ボタンAが押されたとき」ブロックを右側に持ってきて、
「A▼」を押して「ボタンBが押されたとき」に変更しよう

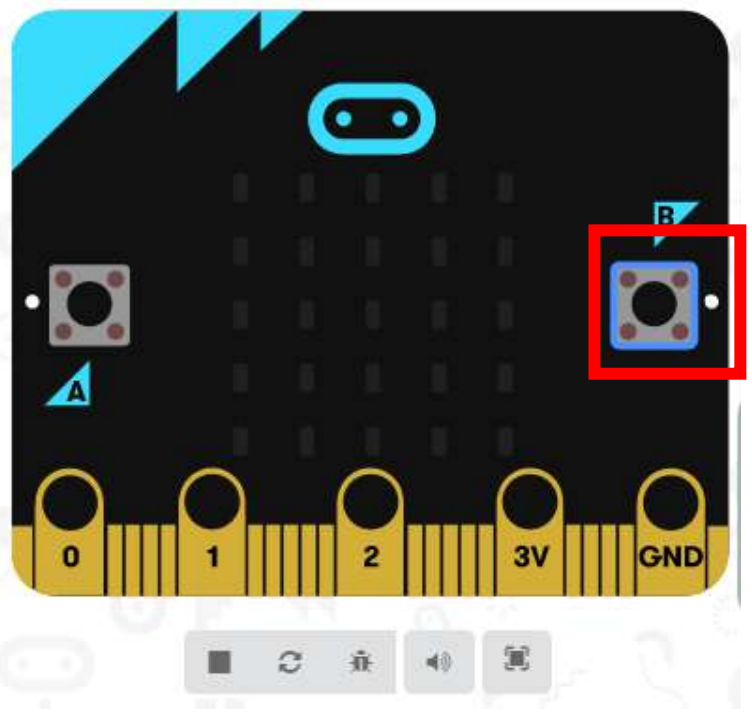
基本操作を学ぼう

基本から「表示を消す」ブロックを持ってきて
「ボタンBが押されたとき」ブロックと組み合わせよう



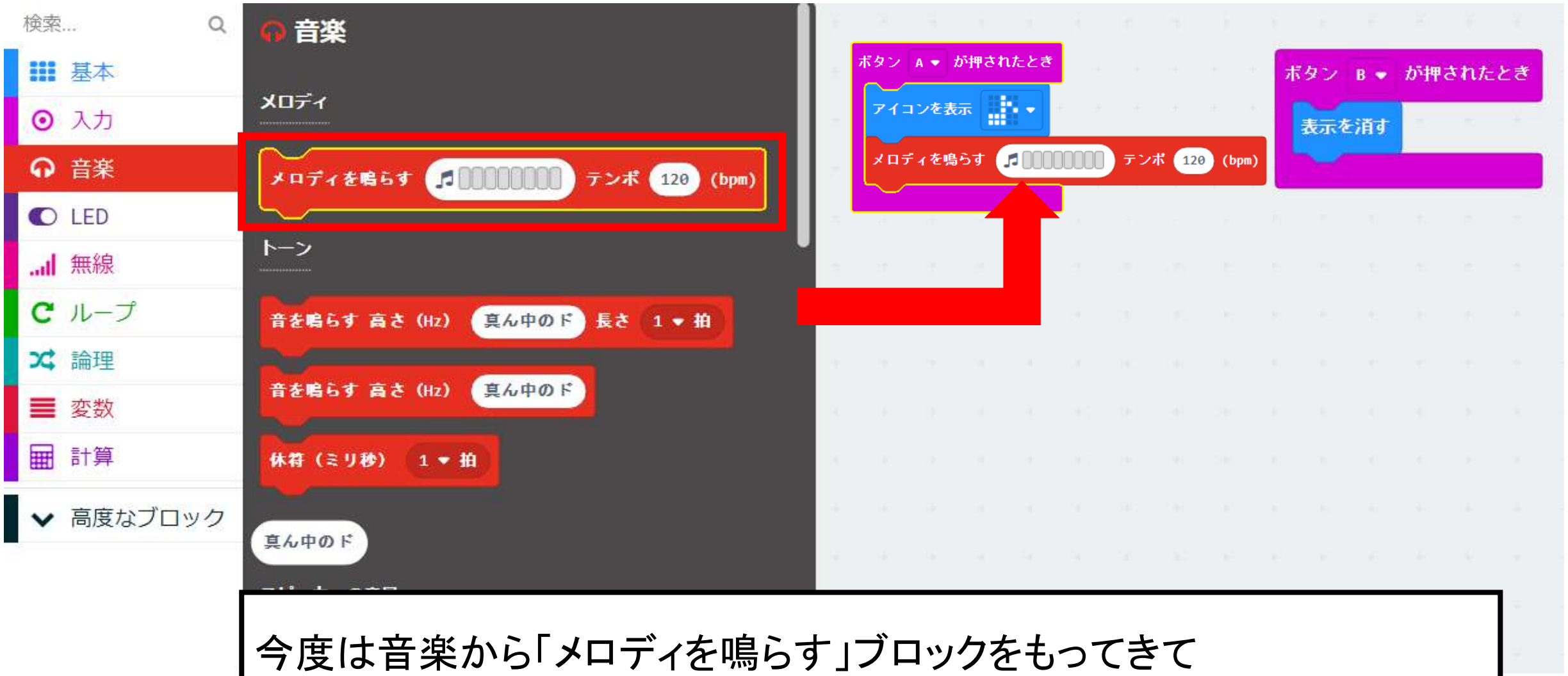
基本操作を学ぼう





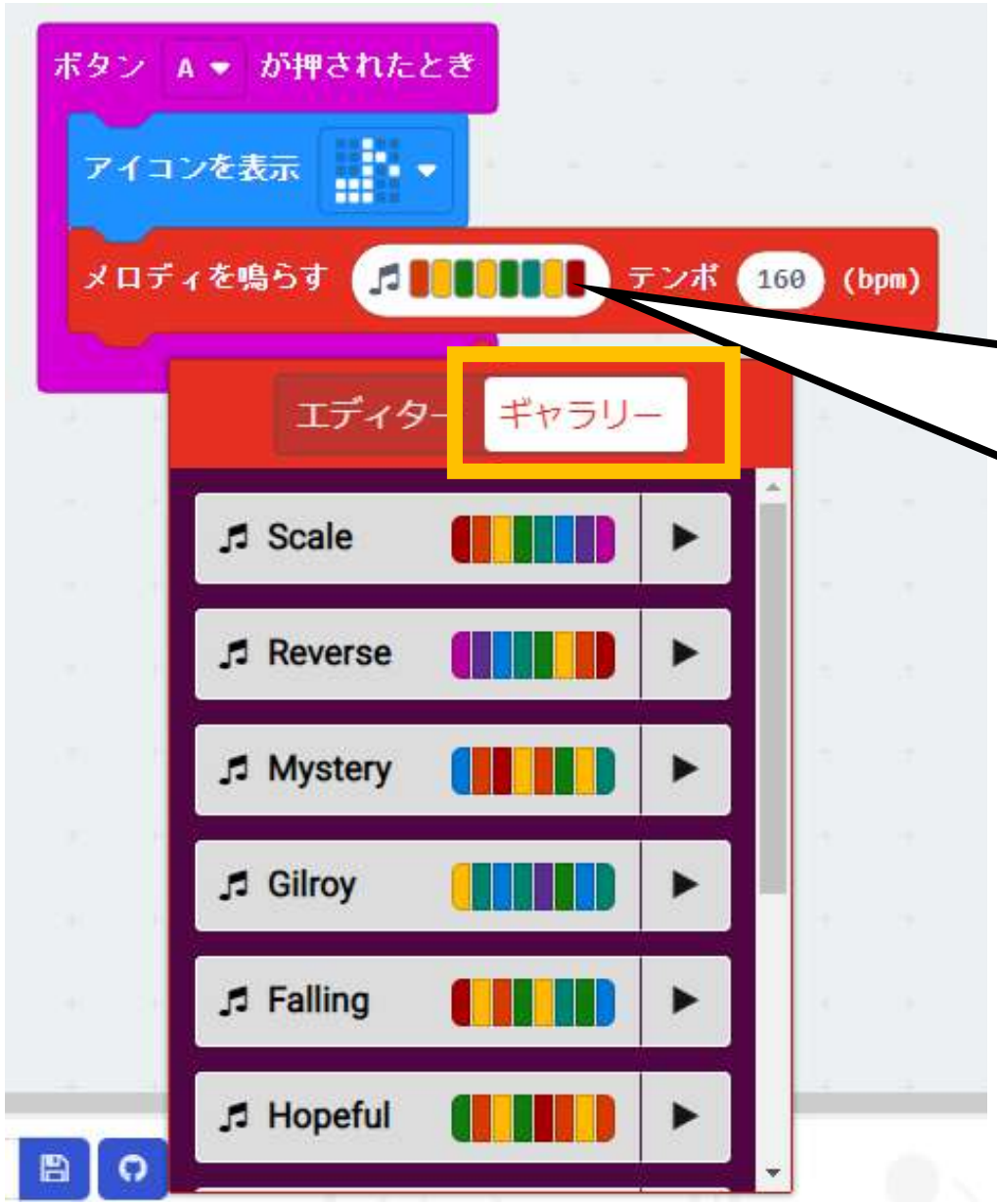
Aボタンを押すとLEDが光ってBボタンで消える仕組みが完成

基本操作を学ぼう



今度は音楽から「メロディを鳴らす」ブロックをもってきて
「アイコンを表示」ブロックの下に組み合わせよう

基本操作を学ぼう



音符マークをクリックして、ギャラリーから好きなメロディを選んでみよう

右側のテンポは数字が大きければ早くなる

基本操作を学ぼう

Micro bitの本体にこのプログラムを入れてみよう！

Micro bitで温度を測ろう

Micro bitの画面から、新しいプロジェクトをクリックしてプログラムを作成する

マイプロジェクト　すべて表示する

空の新しいプロジェクトを作成します。

新しいプロジェクト

探知機生徒用

16時間前

探知機

21時間前

チュートリアル

初めて？ Start Here!

点滅するハート

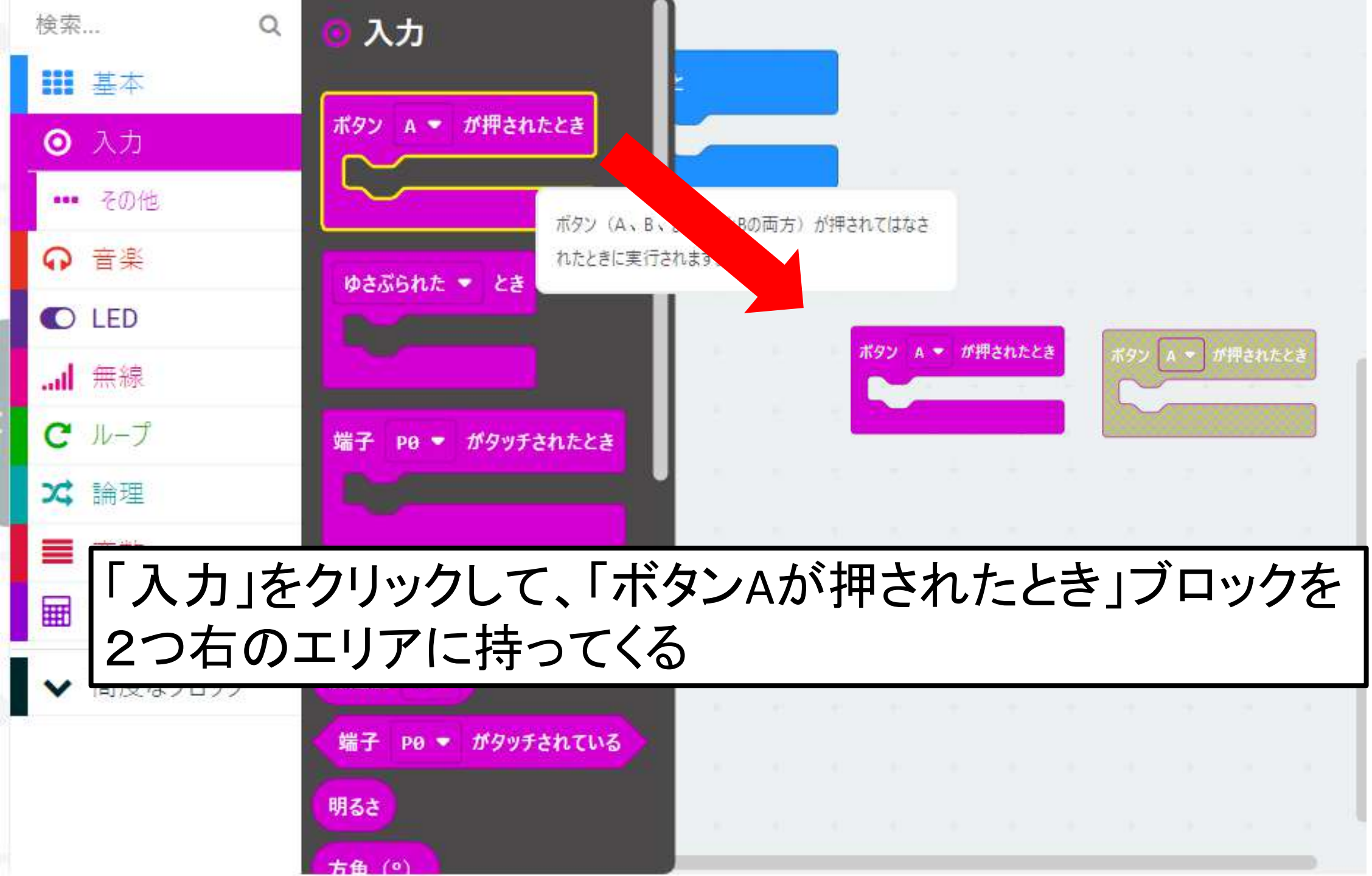
Name Tag

Smiley Buttons

Dice

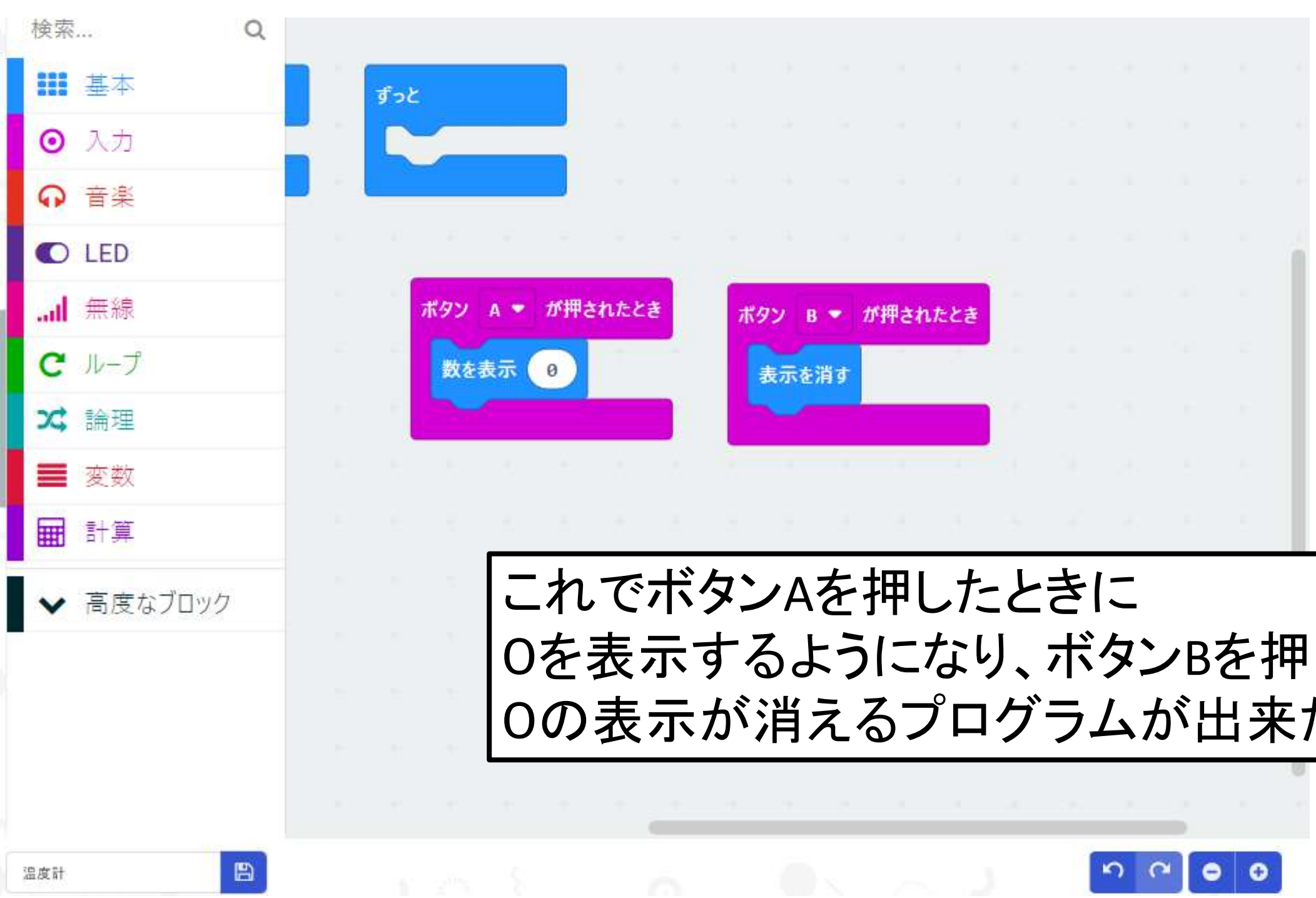
ブロックを配置してプログラムを作る画面が出てくる

ここからボタンを押したときに温度を測る機能を用意する

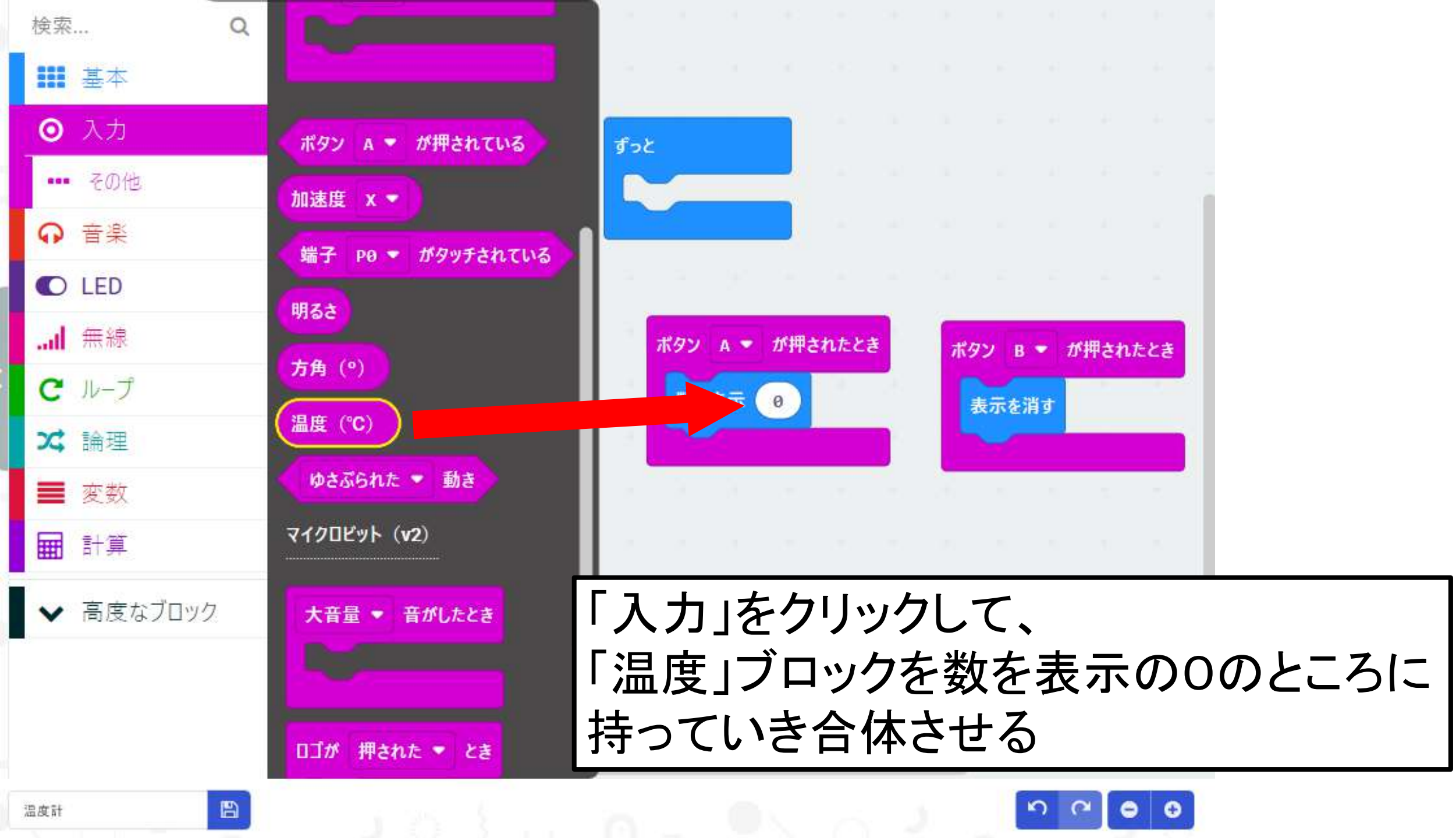

検索...
基本
入力
その他
音楽
LED
無線
ループ
論理
入力
ボタン A が押されたとき
ゆさぶられた とき
端子 P0 がタッチされたとき
端子 P0 がタッチされている
明るさ
ボタン A が押されたとき
ボタン A が押されたとき
「入力」をクリックして、「ボタンAが押されたとき」ブロックを
2つ右のエリアに持ってくる

片方の「ボタンAが押されたとき」ブロックの
▼をクリックして、Bにチェックを付ける


検索...
基本
入力
音楽
LED
無線
ループ
論理
変数
計算
高度なブロック
基本
数を表示 0
画面上のに数字をスクロー ます。数が1桁で、画面
上に収まる場合、スクロールしま
アイコンを表示
文字列を表示 "Hell
表示を消す
ずっと
ずっと
ボタン A が押されたとき
ボタン B が押されたとき
「基本」をクリックして
「数を表示」ブロックをAに
「表示を消す」ブロックをBに合体させる



これでボタンAを押したときに
0を表示するようになり、ボタンBを押したときは
0の表示が消えるプログラムが出来た。


検索...
基本
入力
その他
音楽
LED
無線
ループ
論理
変数
計算
高度なブロック
ボタン A が押されている
加速度 x
端子 P0 がタッチされている
明るさ
方角（°）
温度（℃）
ゆさぶられた 動き
マイクロビット（v2）
大音量 音がしたとき
ロゴが 押された とき
ずっと
ボタン A が押されたとき
0
ボタン B が押されたとき
表示を消す
「入力」をクリックして、
「温度」ブロックを数を表示の0のところに
持っていき合体させる
温度計

これでボタンAを押すとLEDがその場所の温度を表示してくれる

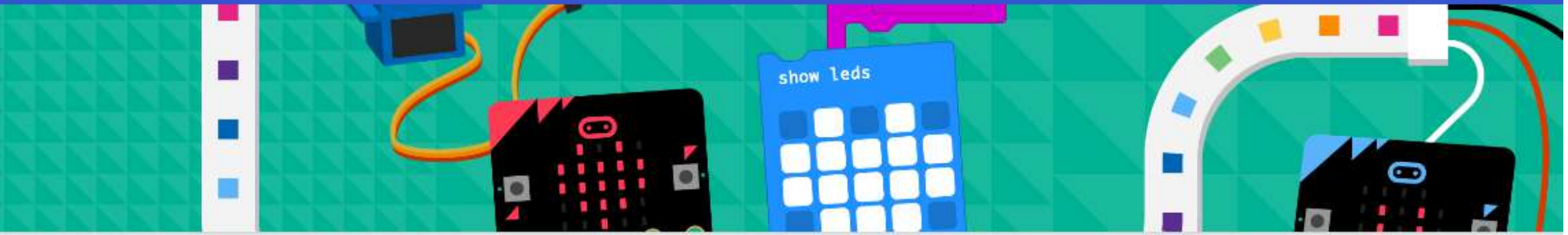
Micro bitをパソコンとつないでダウンロード

実際にプログラムを動かしてみよう！

じゃんけんしてみよう

micro:bit

ホーム



マイプロジェクト　すべて表示する

新しいプロジェクト

温度

5分前

17時間前

22時間前

新しいプロジェクトをクリックして
プログラムを作成していく

チュートリアル

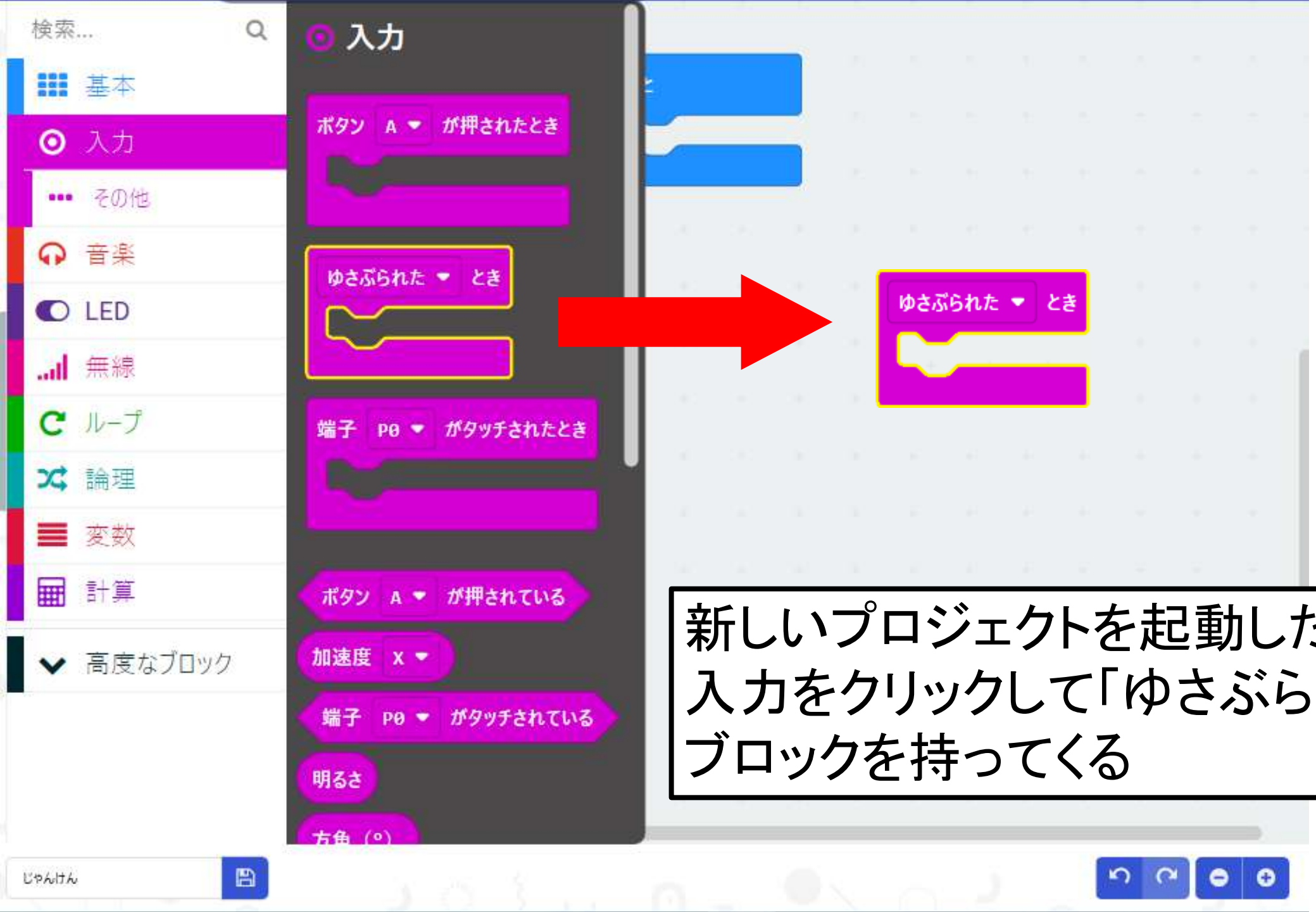
点滅するハート

Name Tag

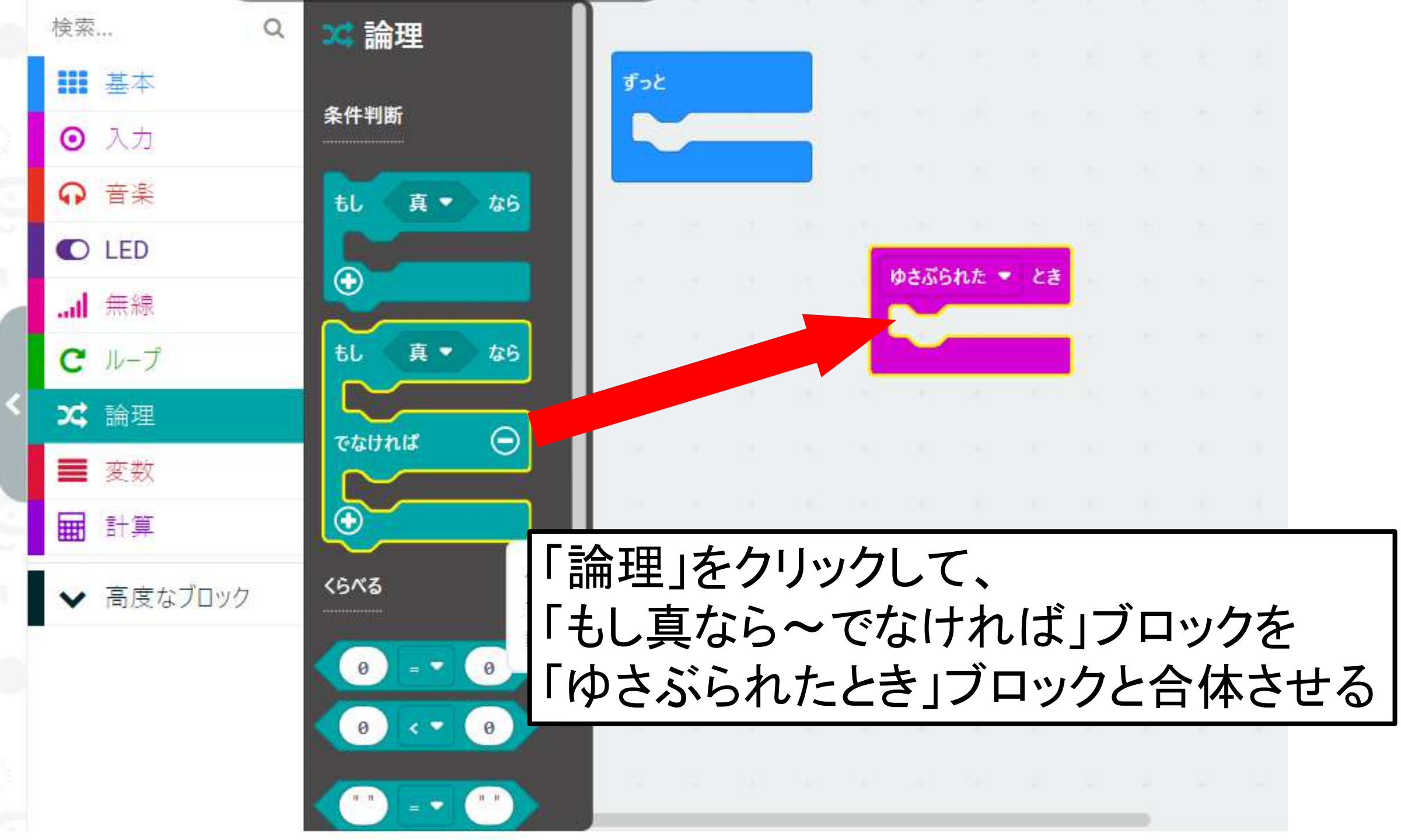
Smiley Buttons

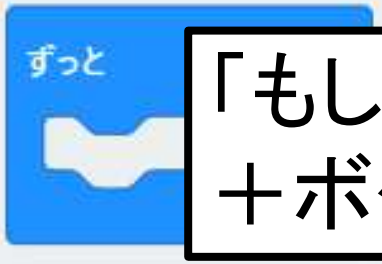
Dice

Love Meter



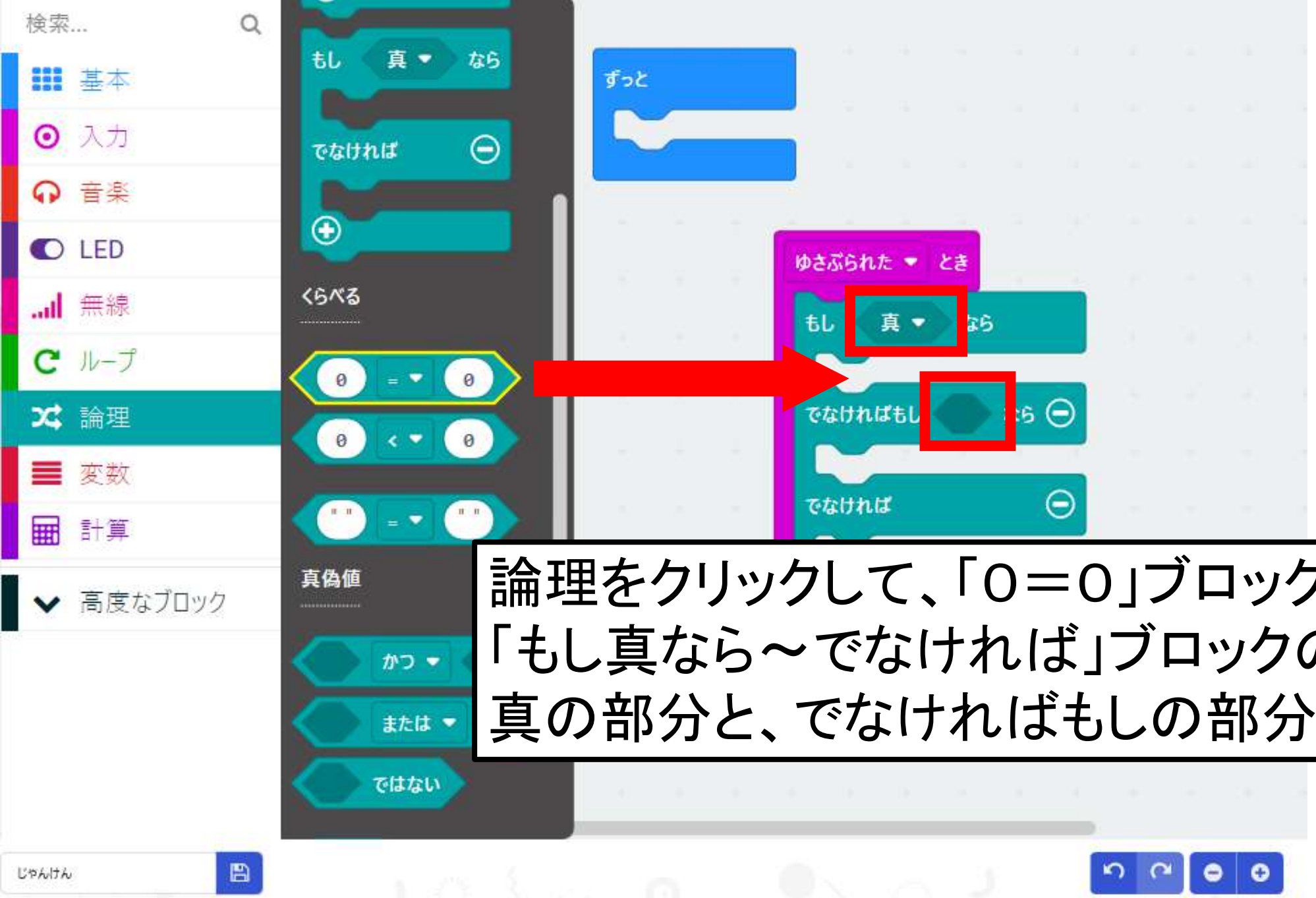
新しいプロジェクトを起動したら、
入力をクリックして「ゆさぶられたとき」
ブロックを持ってくる


検索...
基本
入力
音楽
LED
無線
ループ
論理
変数
計算
高度なブロック
論理
条件判断
もし 真 なら
もし 真 なら
でなければ
くらべる
0 = 0
0 < 0
" " = " "
ずっと
ゆさぶられた ときに
「論理」をクリックして、
「もし真なら～でなければ」ブロックを
「ゆさぶられたとき」ブロックと合体させる



「もし真なら～でなければ」ブロックの
＋ボタンを押して画像のように条件を3つにする

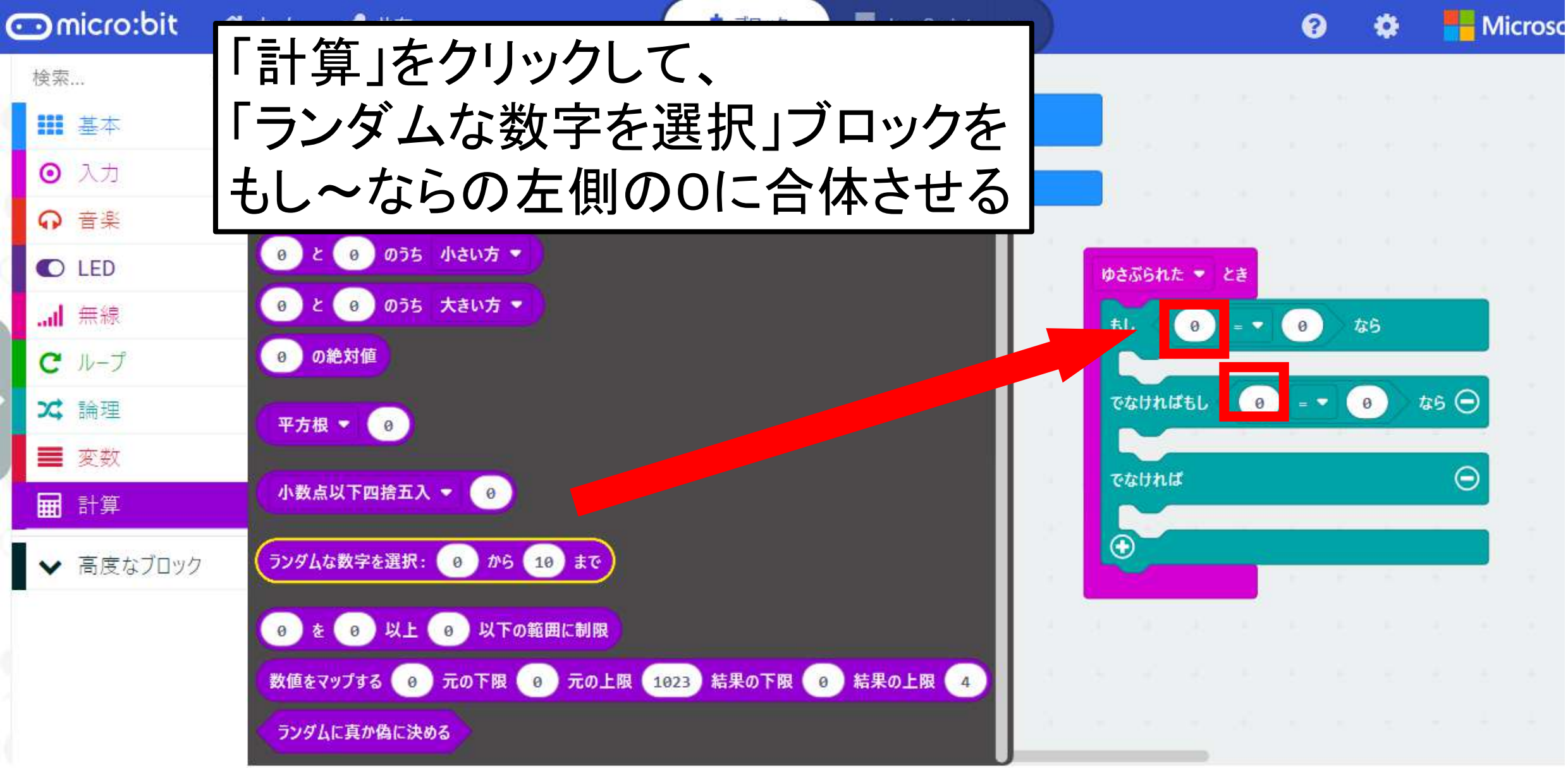
最初の値が真の場合には、最初のかたまりのブロックを動かします。そうでない場合で、二番目の値が真なら、二番目のかたまりのブロックを動かします。どちらの値も真でない場合には、最後のかたまりのブロックを動かします。

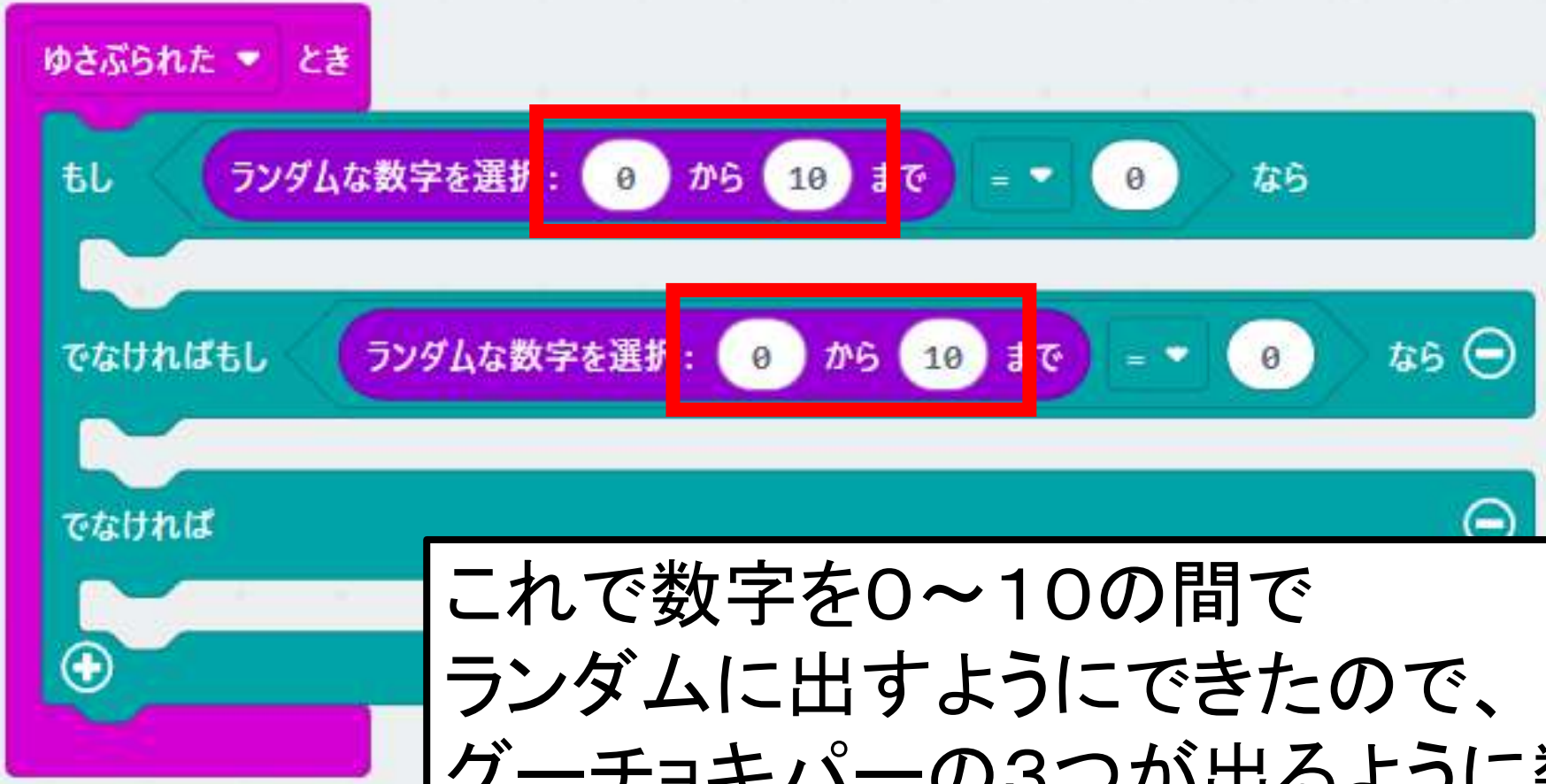


論理をクリックして、「０＝０」ブロックを、
「もし真なら～でなければ」ブロックの
真の部分と、でなければもしの部分に合体させる

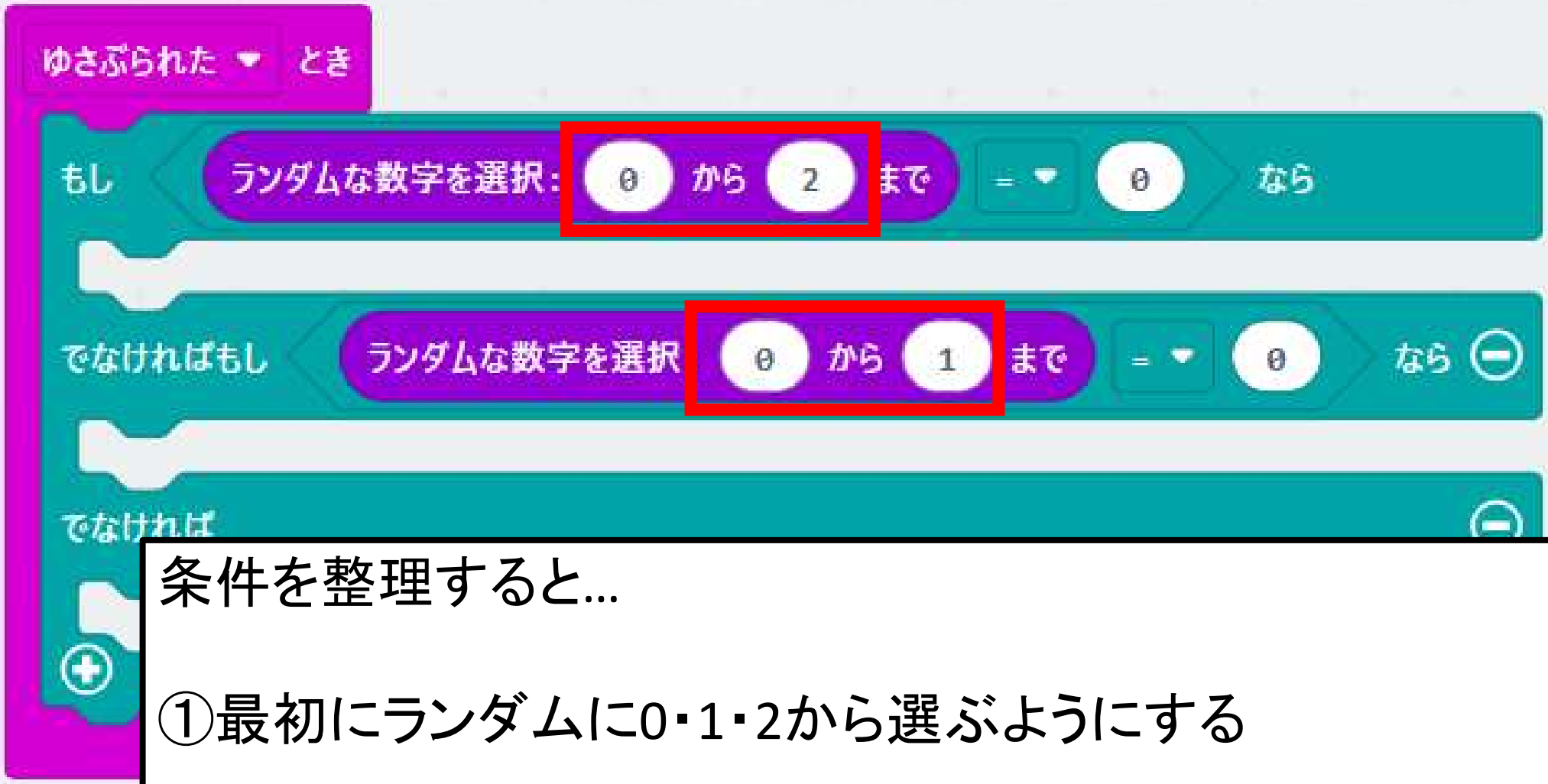
ここから条件の分岐とLEDの表示を設定していく

「計算」をクリックして、
「ランダムな数字を選択」ブロックを
もし～ならの左側の0に合体させる





これで数字を0～10の間で
ランダムに出すようにできたので、
グーチョキパーの3つが出るように数字を変更する

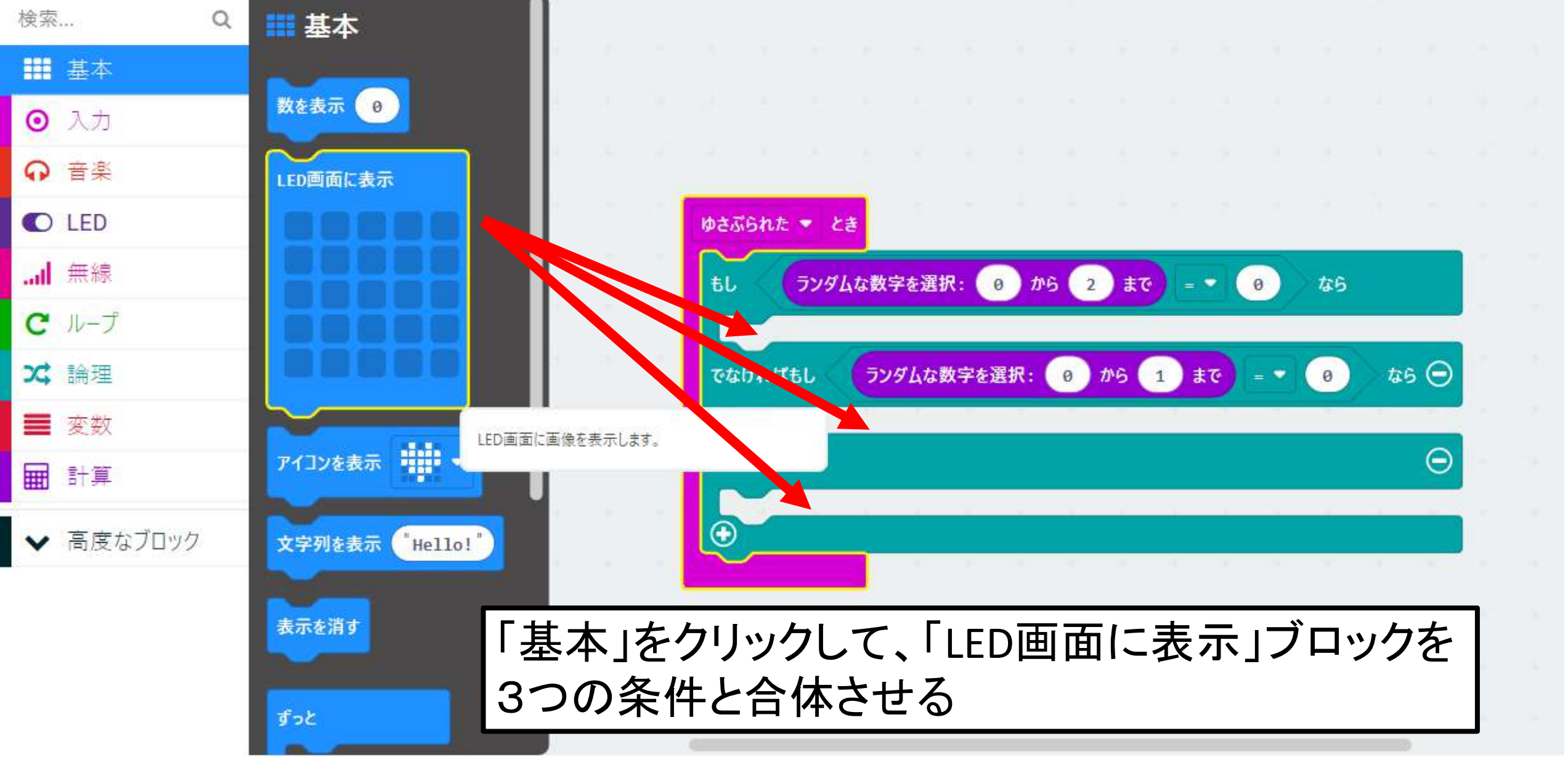


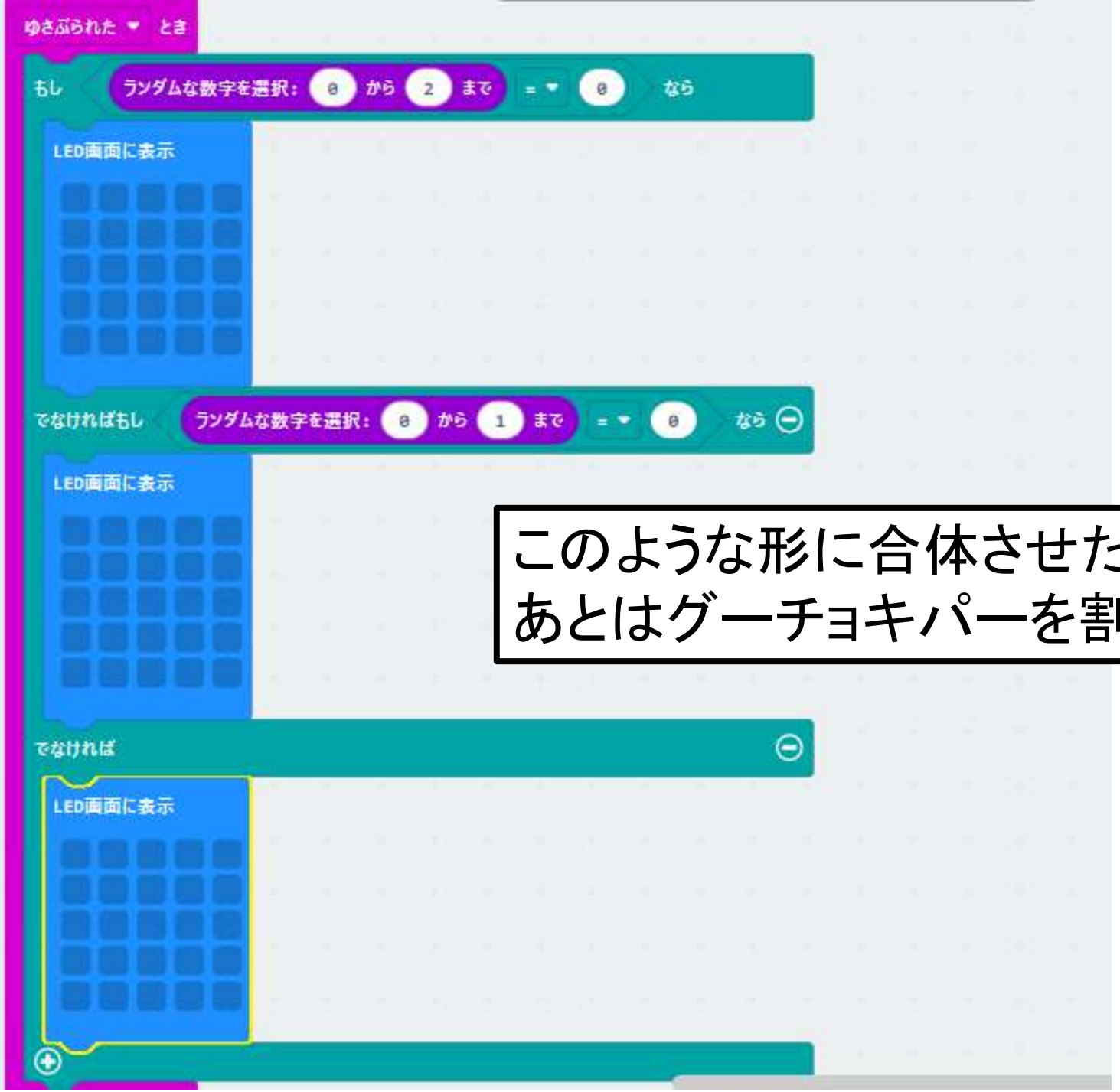
条件を整理すると…

①最初にランダムに0・1・2から選ぶようにする

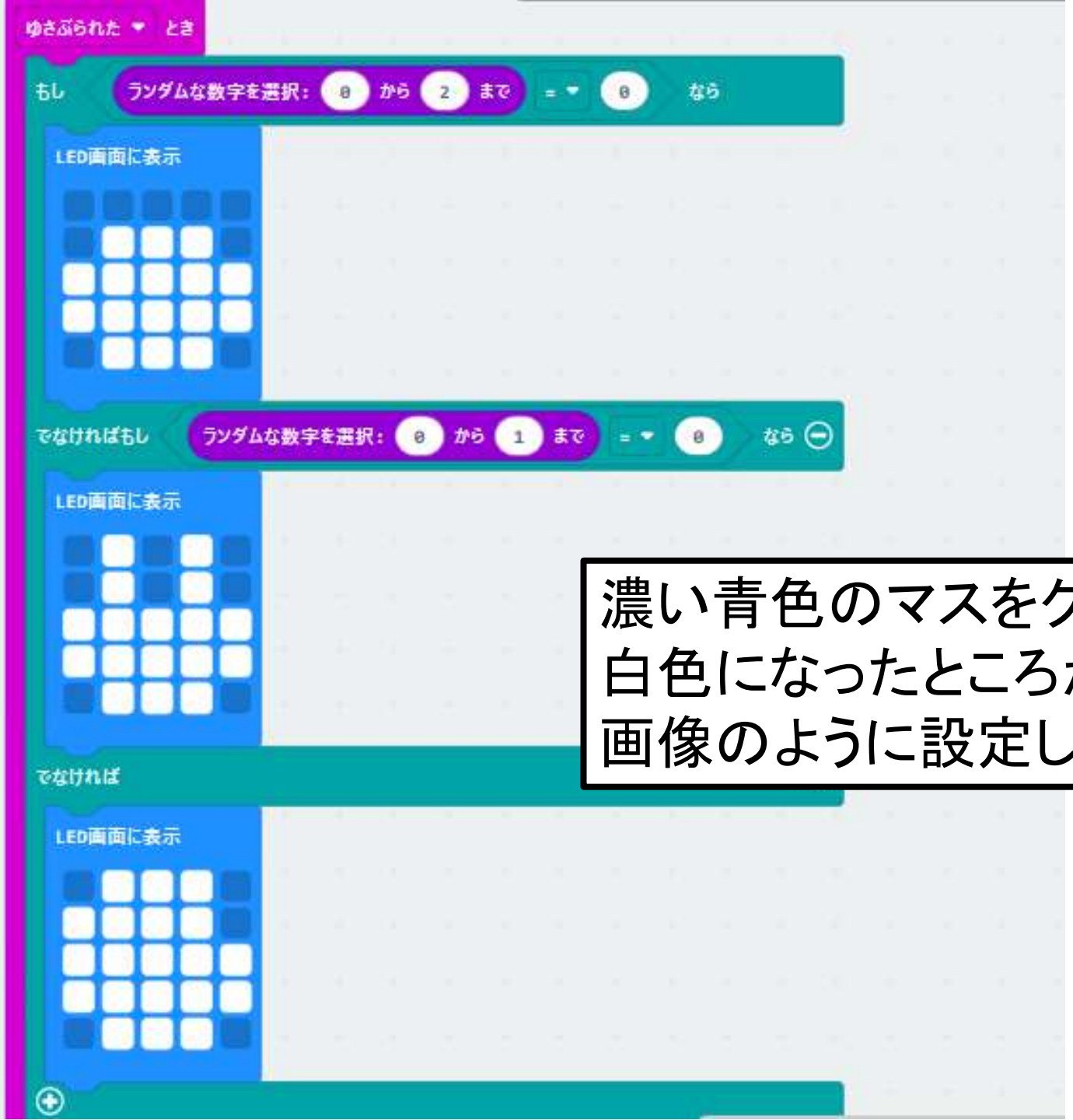
②0が選ばれたら結果を出し、そうでないときは0・1のどちらかを選ぶ

③2回目の条件で1が選ばれたら対応する結果を表示する


検索...
基本
入力
音楽
LED
無線
ループ
論理
変数
計算
高度なブロック
基本
数を表示 0
LED画面に表示
アイコンを表示
LED画面に画像を表示します。
文字列を表示 "Hello!"
表示を消す
ずっと
ゆさぶられた とき
もし ランダムな数字を選択: 0 から 2 まで = 0 なら
でなければもし ランダムな数字を選択: 0 から 1 まで = 0 なら
「基本」をクリックして、「LED画面に表示」ブロックを
3つの条件と合体させる



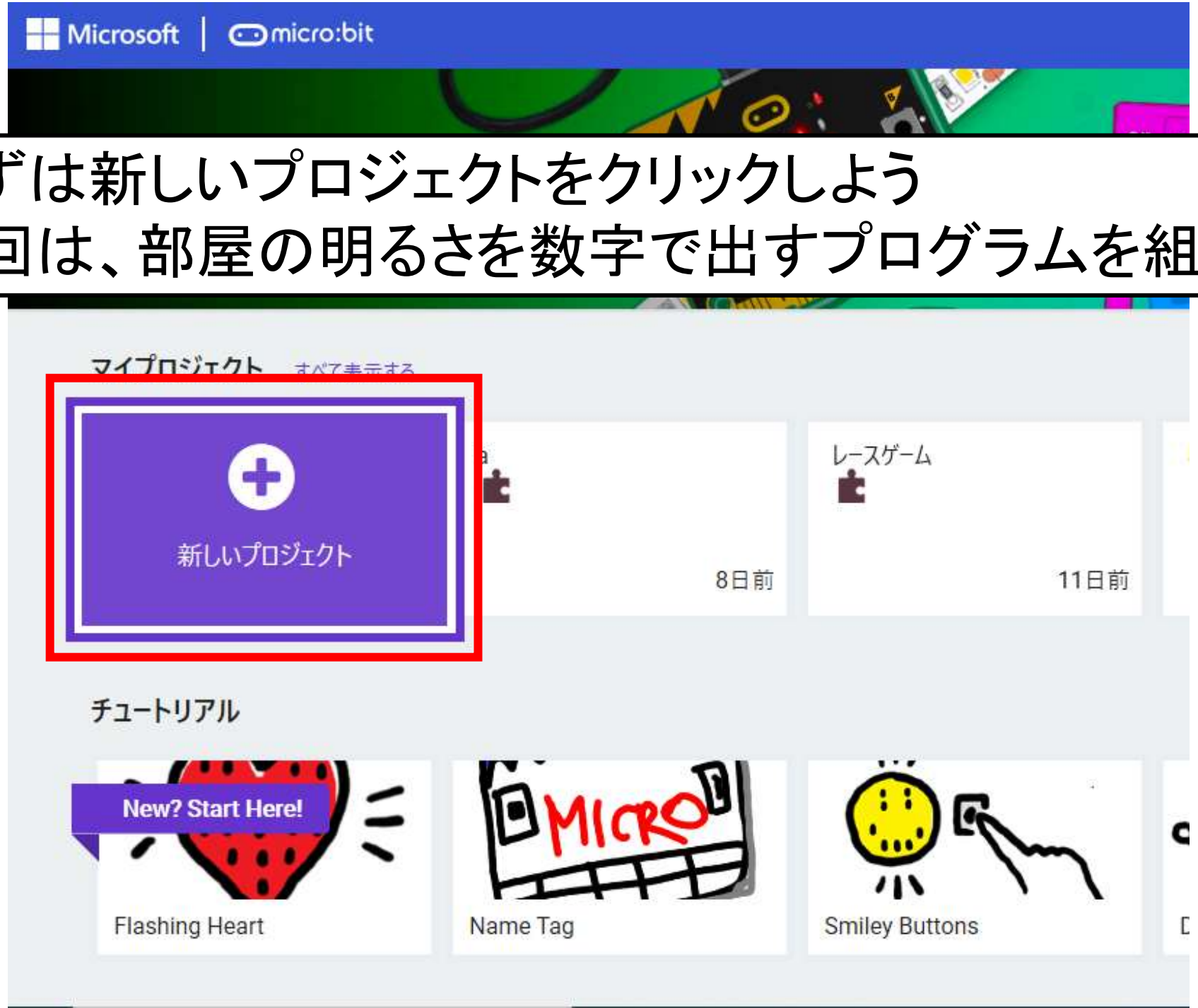
このような形に合体させたら、
あとはグーチョキパーを割り当てる



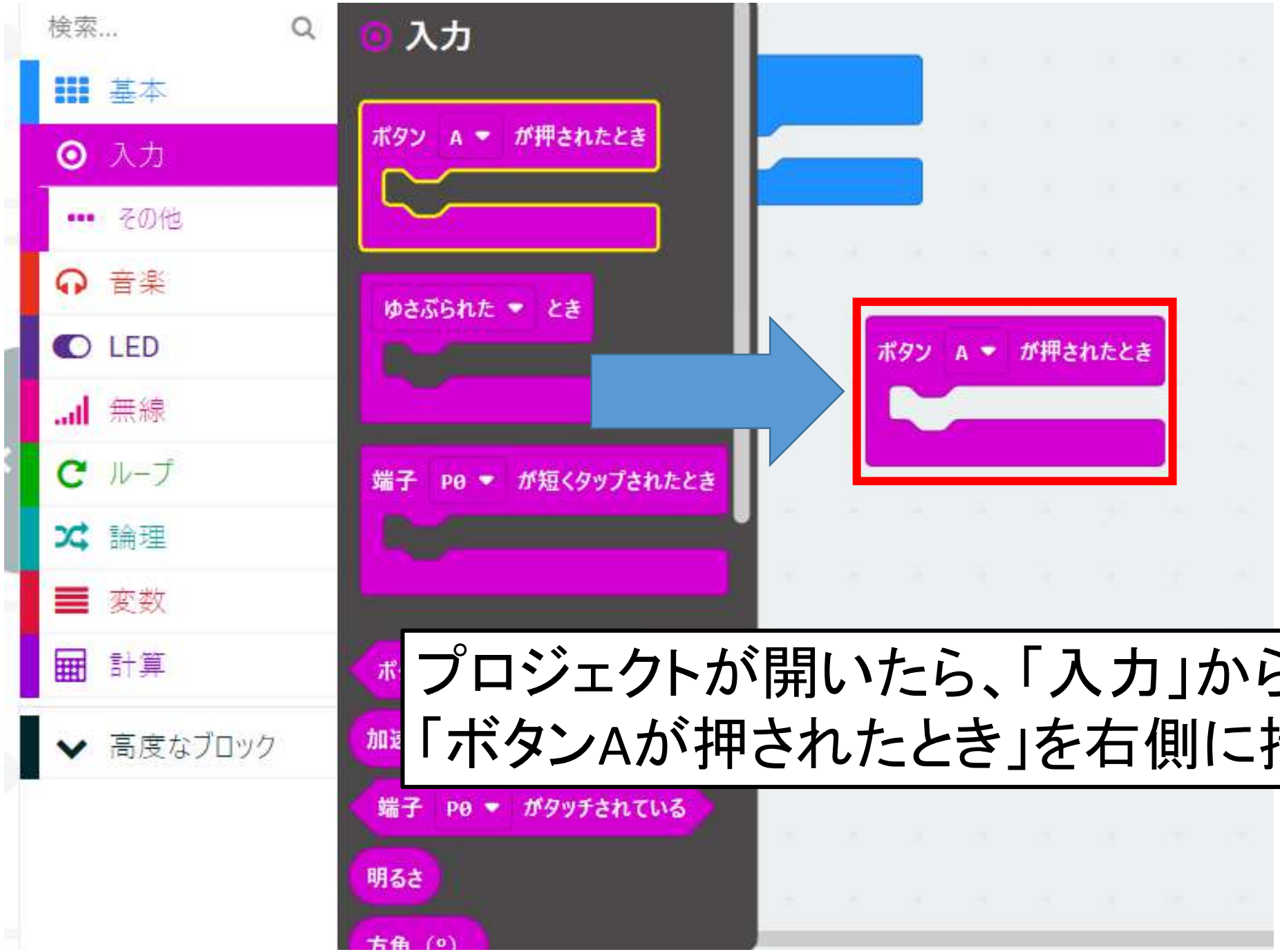
濃い青色のマスをクリックして
白色になったところが実際に光るので
画像のように設定してみよう

PCとMicro bitをつないで
プログラムを入れてみよう！

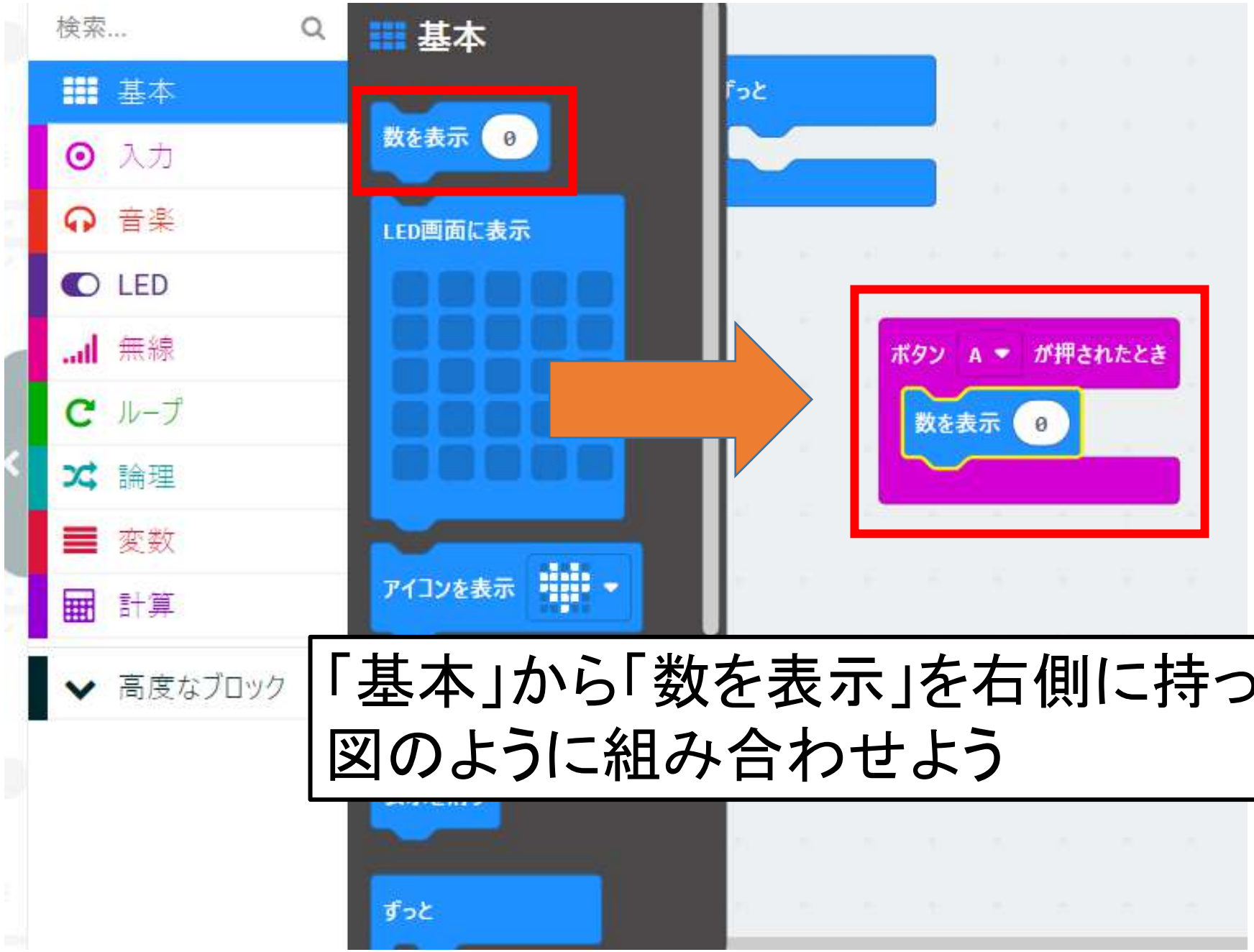
明るさセンサーをつくろう


Microsoft | micro:bit
まずは新しいプロジェクトをクリックしよう
今回は、部屋の明るさを数字で出すプログラムを組むよ
マイプロジェクト
すべて表示する
新しいプロジェクト
レースゲーム
8日前
11日前
チュートリアル
New? Start Here!
Flashing Heart
Name Tag
Smiley Buttons

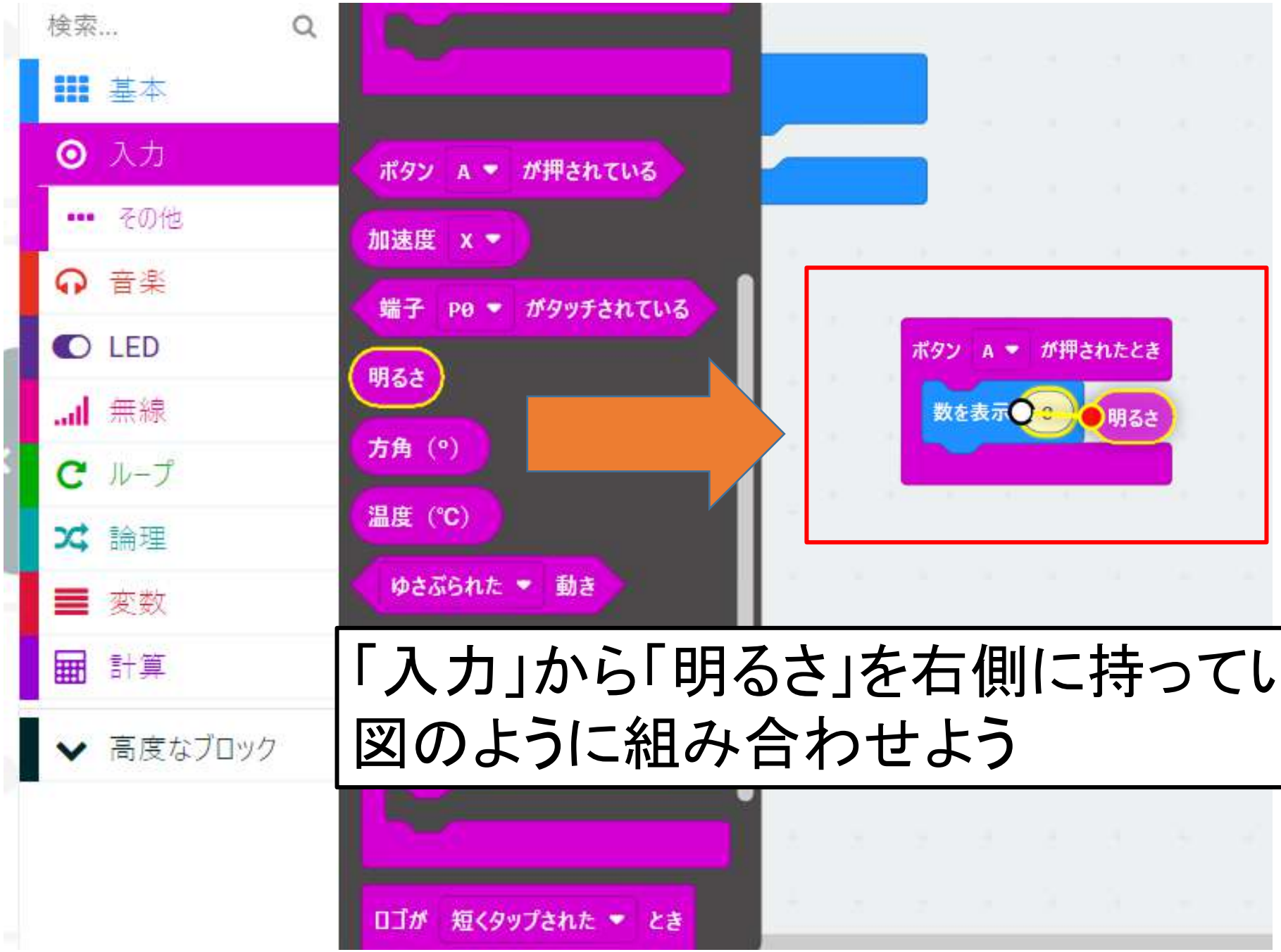
プロジェクトの名前を決めよう



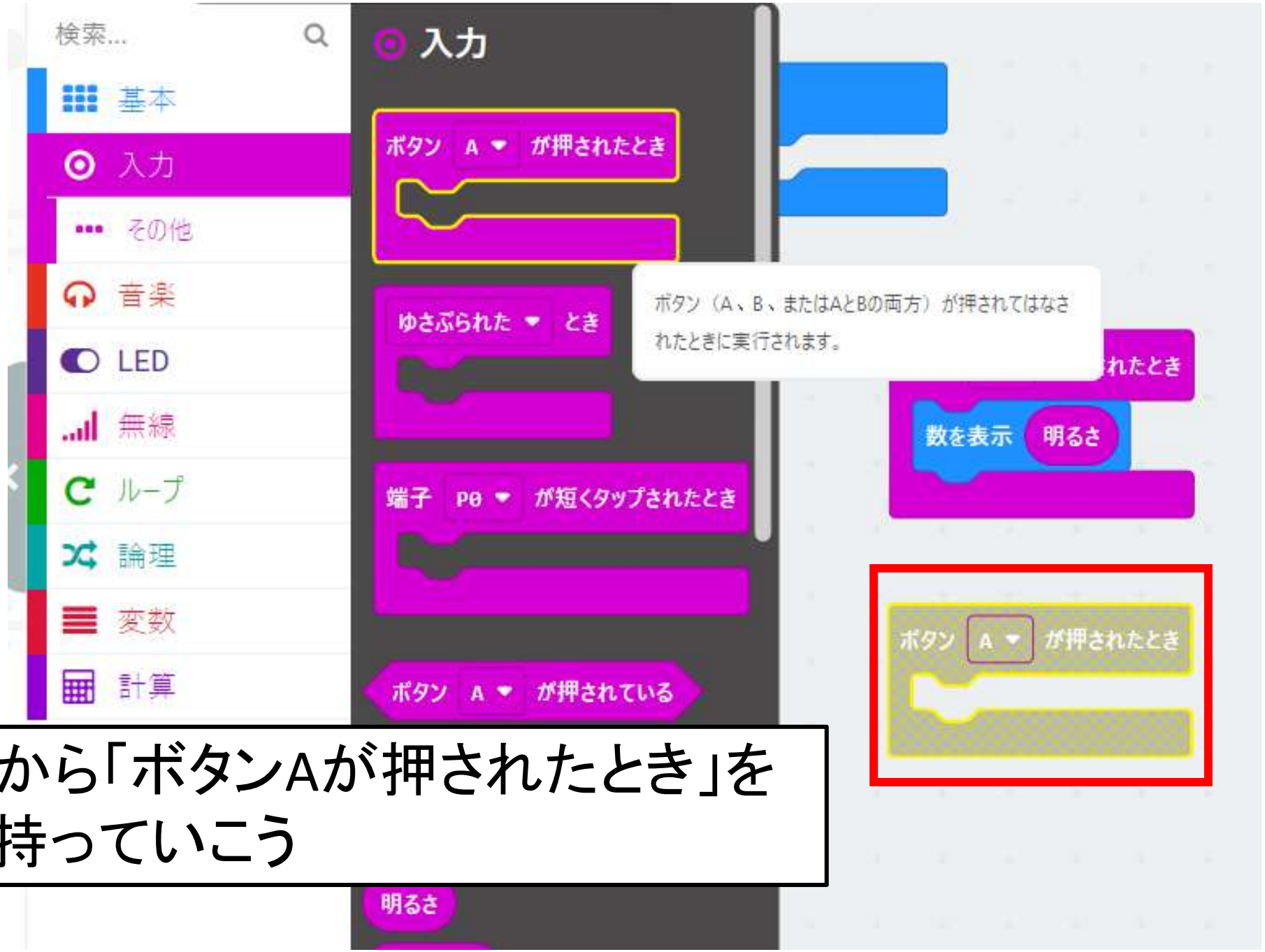
プロジェクトが開いたら、「入力」から
「ボタンAが押されたとき」を右側に持っていこう



「基本」から「数を表示」を右側に持っていって
図のように組み合わせよう



「入力」から「明るさ」を右側に持っていって
図のように組み合わせよう

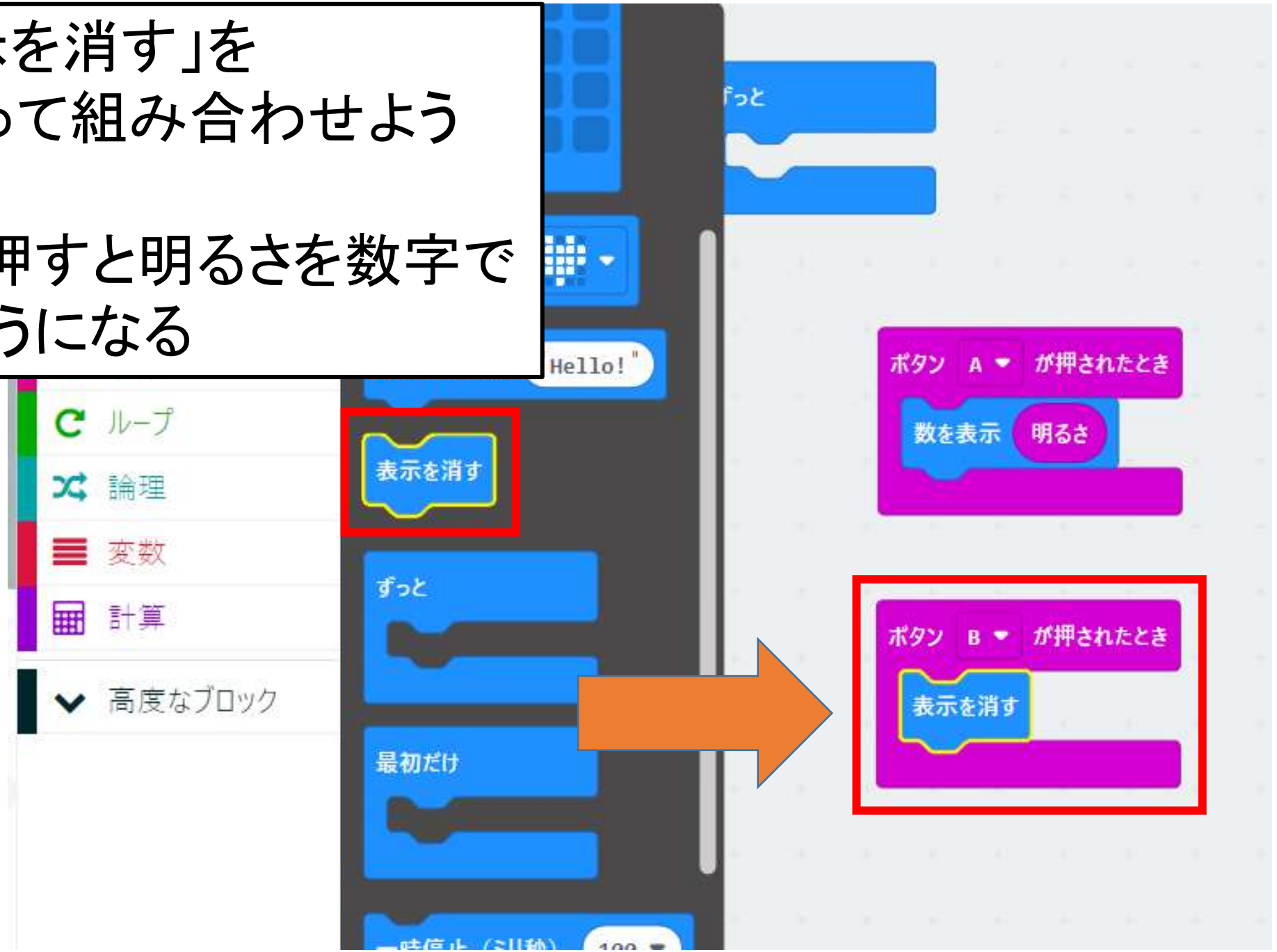


「入力」から「ボタンAが押されたとき」を右側に持っていこう

「ボタンAが押されたとき」をクリックして
「ボタンBが押されたとき」に変更しよう

「基本」から「表示を消す」を
右側に持っていって組み合わせよう

これでAボタンを押すと明るさを数字で
表示してくれるようになる

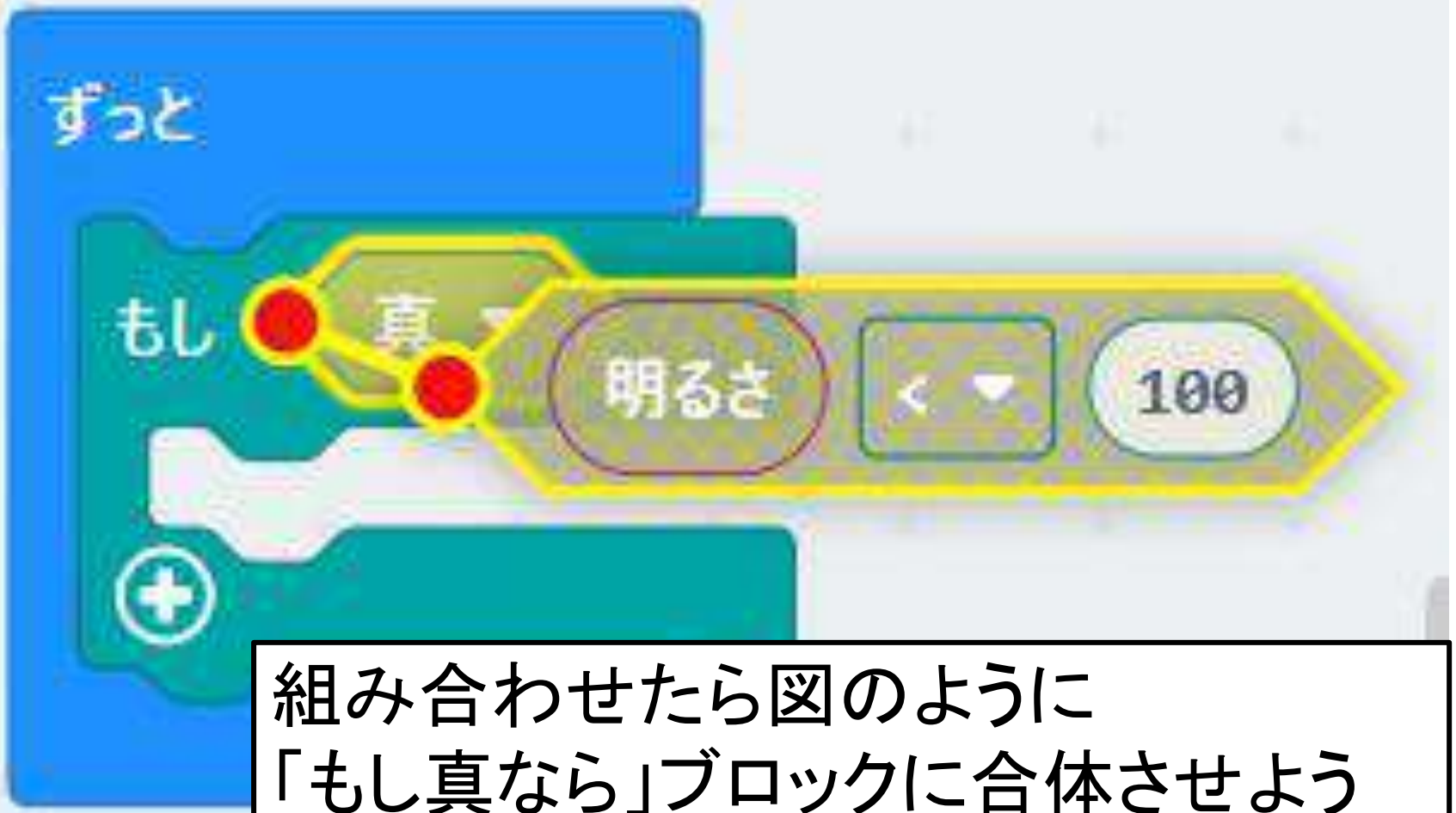



検索...
基本
入力
音楽
LED
無線
ループ
論理
変数
計算
高度なブロック
論理
条件判断
もし 真 なら
もし 真 なら
でなければ
くらべる
0 =
A が押されたとき
を表示 明るさ
B が押されたとき
示を消す
ずっと
もし 真 なら
「論理」から「もし真なら」を
右側に持っていって、最初からある
「ずっと」ブロックに組み合わせよう


検索...
基本
入力
音楽
LED
無線
ループ
論理
変数
計算
高度なブロック
でなければ
くらべる
0 = 0
0 < 0
" " = " "
真偽値
かつ
または
ではない
A が押されたとき
を表示 明るさ
B
示を消す
ずっと
もし 真 なら
0 < 0
つづけて「論理」から「0＝0」を
右側に持っていこう

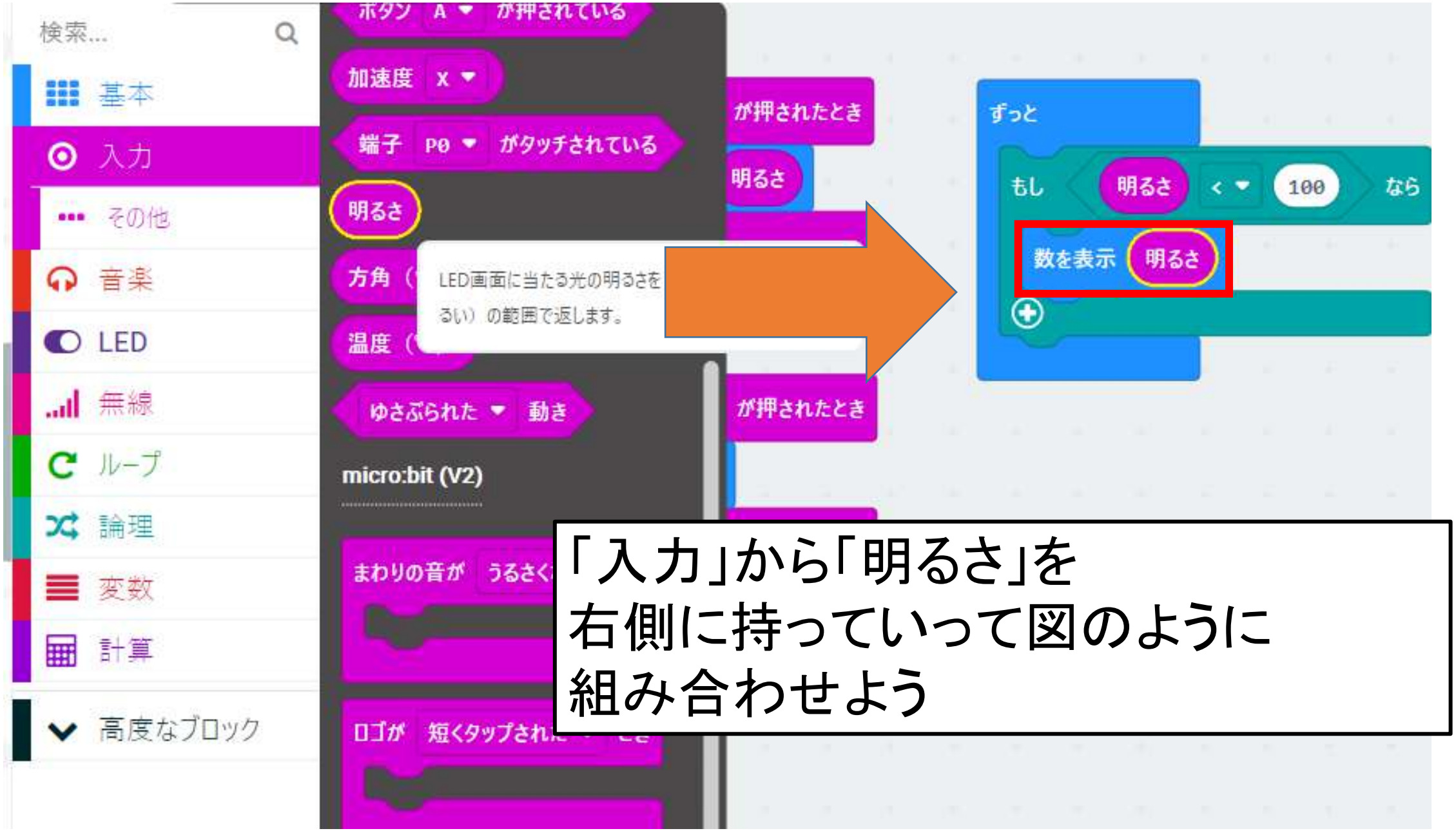

検索...
基本
入力
その他
音楽
LED
無線
ループ
論理
変数
計算
高度なブロック
が押されたとき
明るさ
ボタン A が押されている
加速度 X
端子 P0 がタッチされている
明るさ
方角（°）
温度（°C）
ゆさぶられた
micro:bit (V2)
まわりの音が
ずっと
もし 真 なら
明るさ < 0
「入力」から「明るさ」を
右側に持っていって「0＝0」に
組み合わせよう

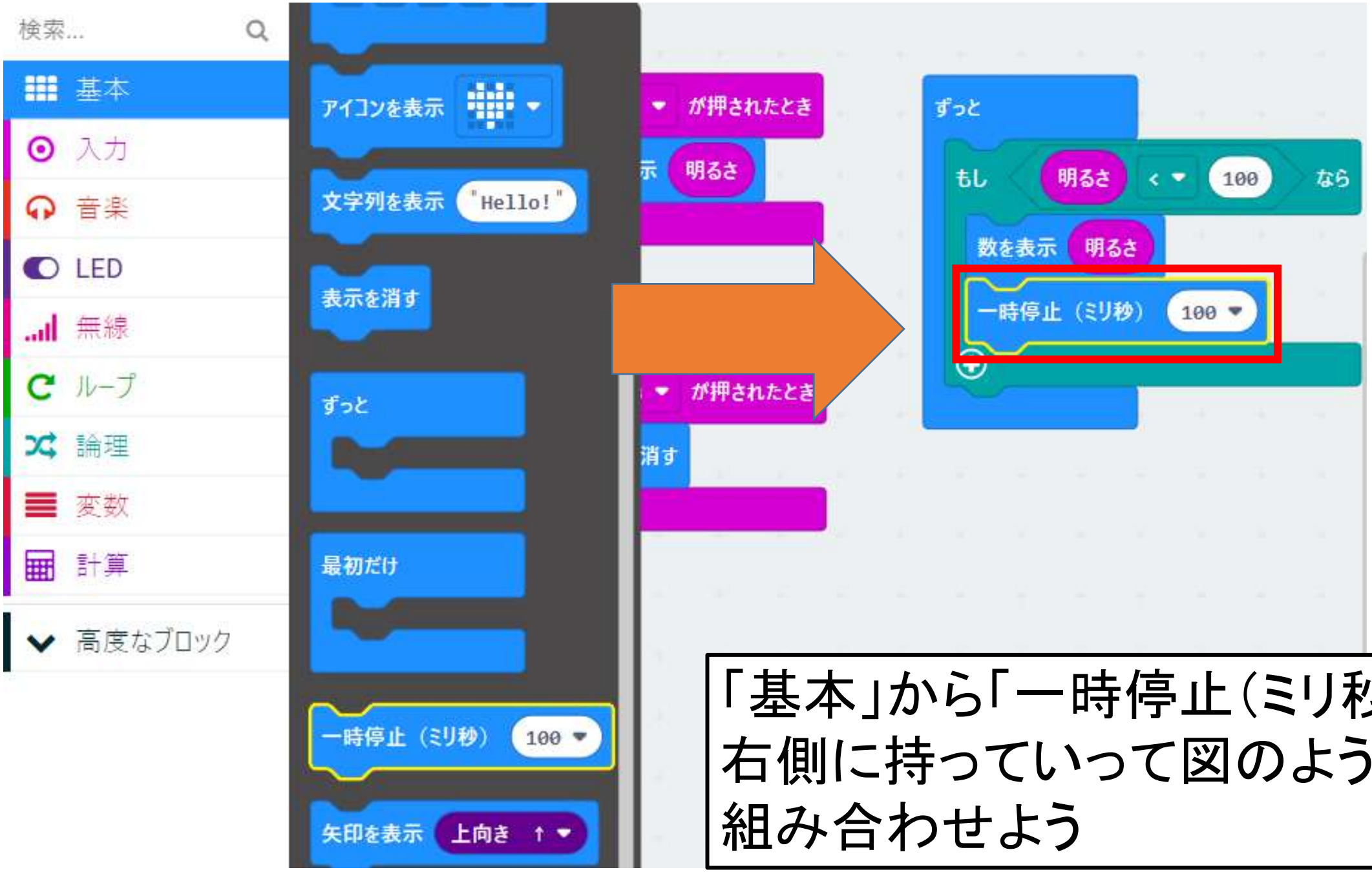
組み合わせたブロックの右側を「１００」に変更しよう



組み合わせたら図のように
「もし真なら」ブロックに合体させよう

「基本」から「数を表示」を
右側に持っていって図のように
組み合わせよう


検索...
基本
入力
その他
音楽
LED
無線
ループ
論理
変数
計算
高度なブロック
ボタン A が押されている
加速度 X
端子 P0 がタッチされている
明るさ
方角（
温度（
LED画面に当たる光の明るさを
るい）の範囲で返します。
ゆさぶられた 動き
micro:bit (V2)
まわりの音が
ロゴが
が押されたとき
明るさ
が押されたとき
ずっと
もし 明るさ < 100 なら
数を表示 明るさ
「入力」から「明るさ」を
右側に持っていって図のように
組み合わせよう

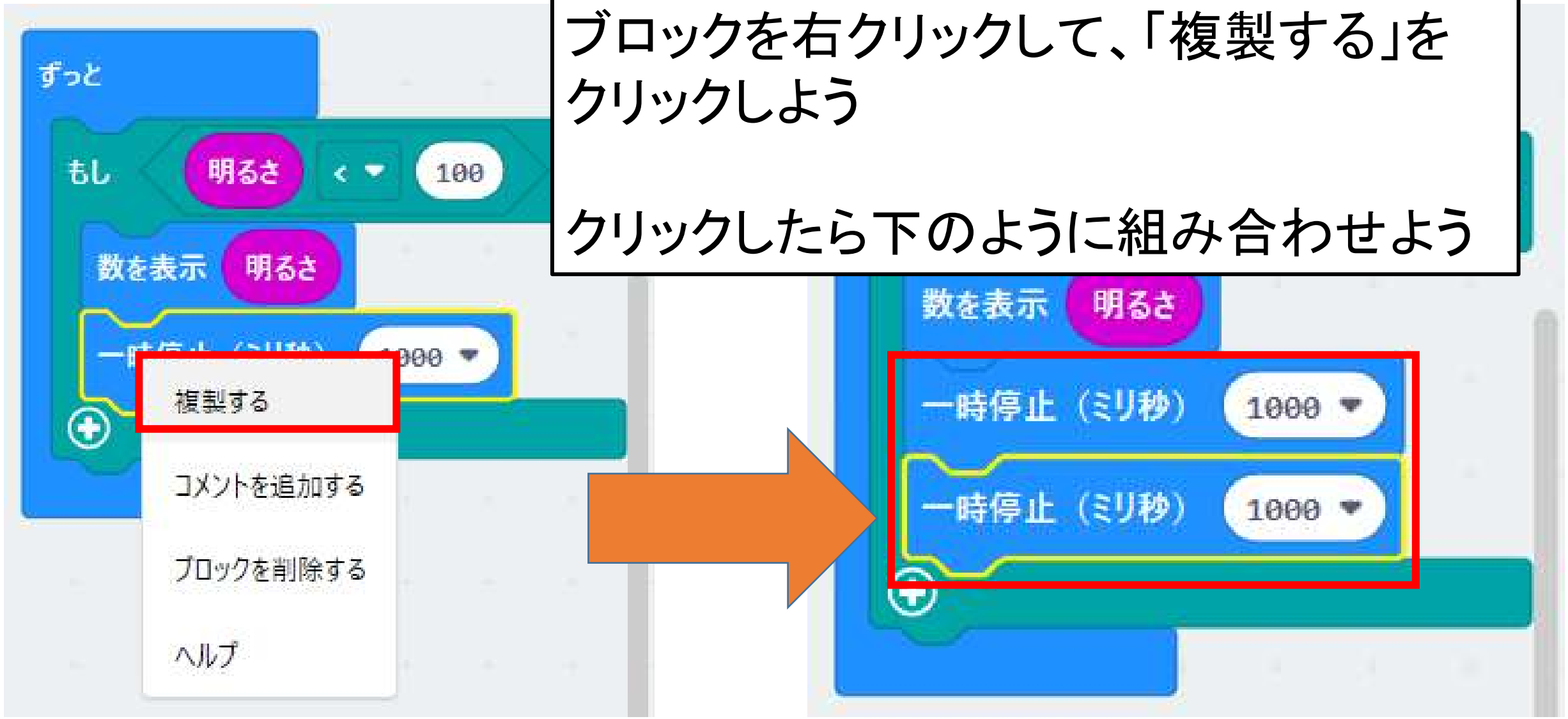


「基本」から「一時停止（ミリ秒）」を
右側に持っていって図のように
組み合わせよう

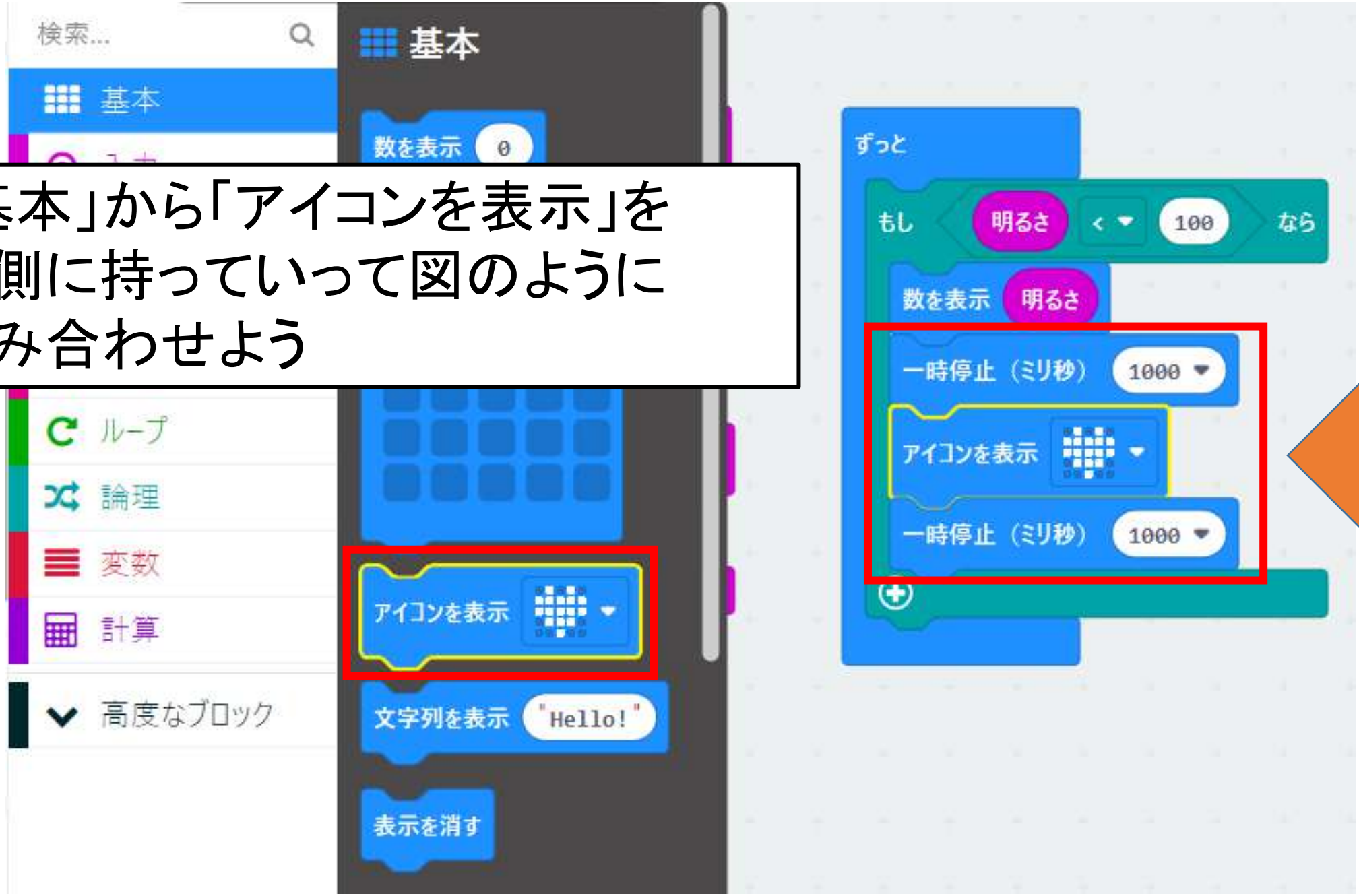
「一時停止（ミリ秒）」の数字を
クリックして1second（1秒）に
設定しよう

設定した「一時停止（ミリ秒）」の
ブロックを右クリックして、「複製する」を
クリックしよう

クリックしたら下のように組み合わせよう



「基本」から「アイコンを表示」を
右側に持っていって図のように
組み合わせよう



アイコンのハートマークをクリックして、
好きなアイコンに変えてみよう

部屋の電気を消したり、机の下に置いて
みたりして、明るさの数字がどれくらい
変わるか見てみよう！

宝探しゲームを作ろう

micro:bit

ホーム

空の新しいプロジェクトを作成します。

新しいプロジェクト

Microbitの画面から、新しいプロジェクトをクリック

20分前

32分前

チュートリアル

点滅するハート

Name Tag

Smiley Buttons

Dice

MakeCode for micro:bit

micro:bit　ホーム　共有　ブロック　JavaScript　Microsoft

検索...

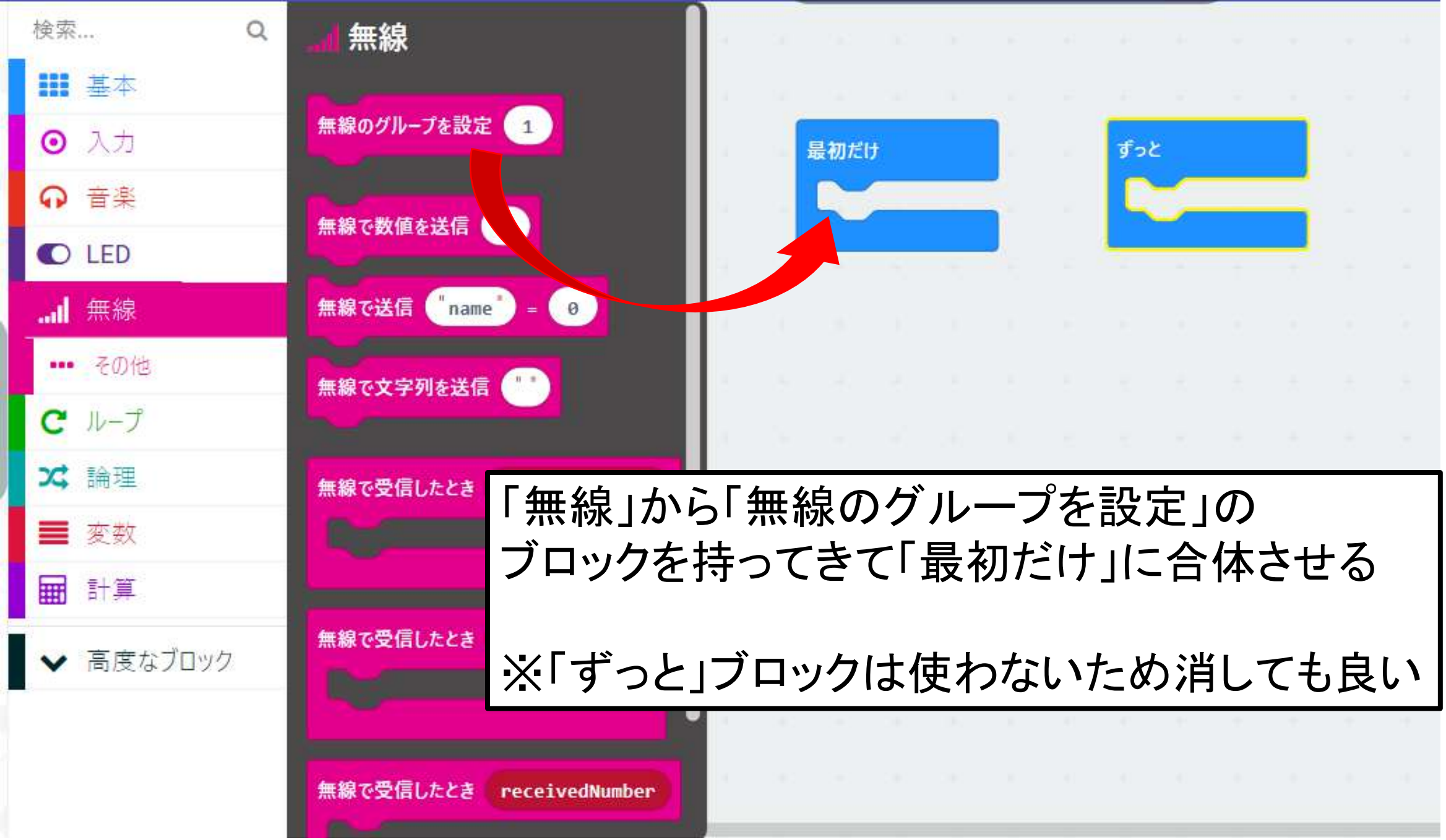
- 基本
- 入力
- 音楽
- LED
- 無線
- ループ
- 論理
- 変数
- 計算
- 高度なブロック

各ブロックを配置してプログラムを
作る画面に移動する。

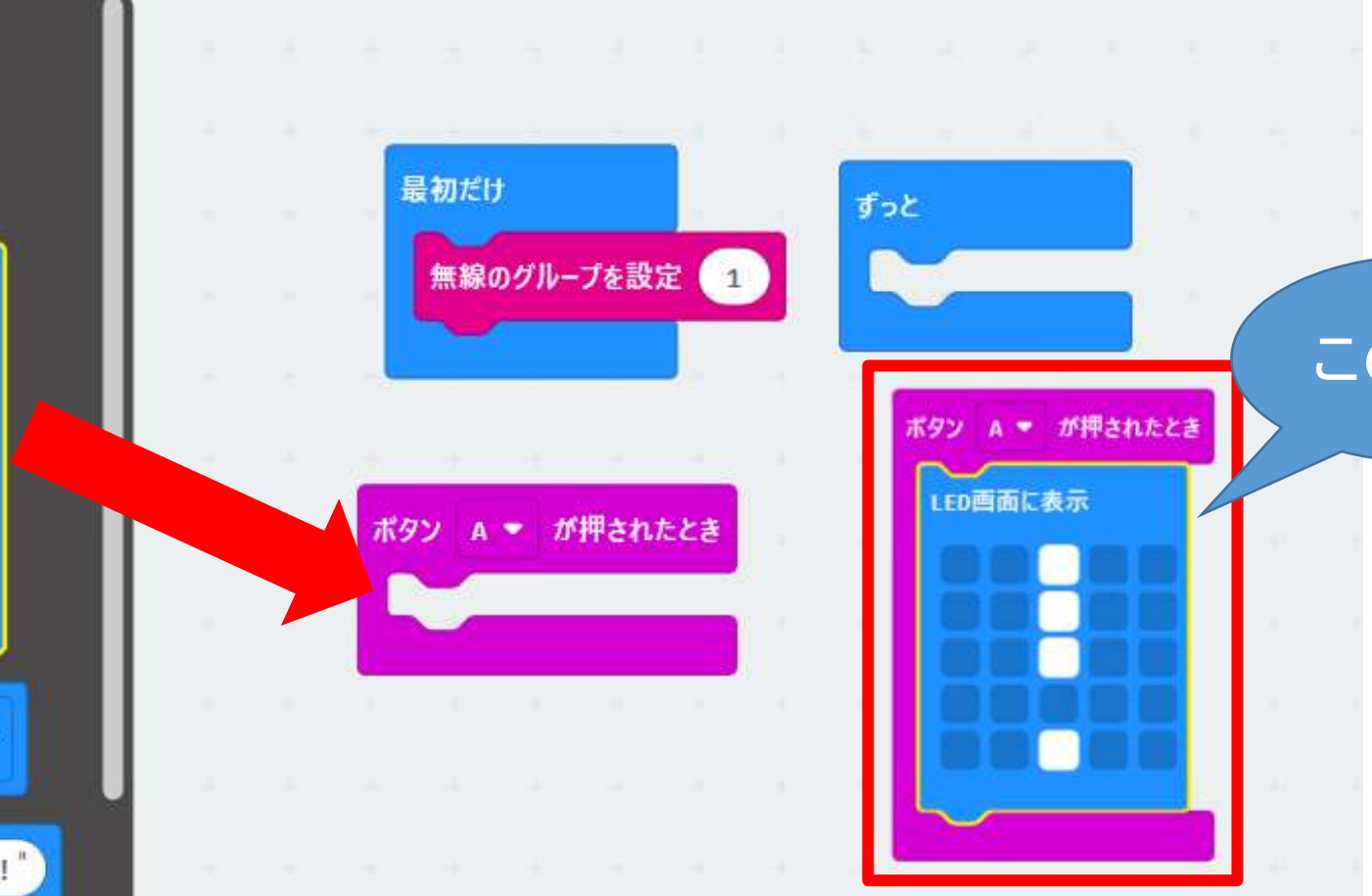
まずは隠すほうの送信機のプログラムを
作成する。

ダウンロード　採知機生徒用

ここに入力して検索　13:11　2021/04/26


検索...
基本
入力
音楽
LED
無線
その他
ループ
論理
変数
計算
高度なブロック
無線
無線のグループを設定 1
無線で数値を送信
無線で送信 "name" = 0
無線で文字列を送信 ""
無線で受信したとき
無線で受信したとき
無線で受信したとき receivedNumber
最初だけ
ずっと
「無線」から「無線のグループを設定」の
ブロックを持ってきて「最初だけ」に合体させる
※「ずっと」ブロックは使わないため消しても良い

「入力」から「ボタンAが押されたとき」のブロックを持ってくる

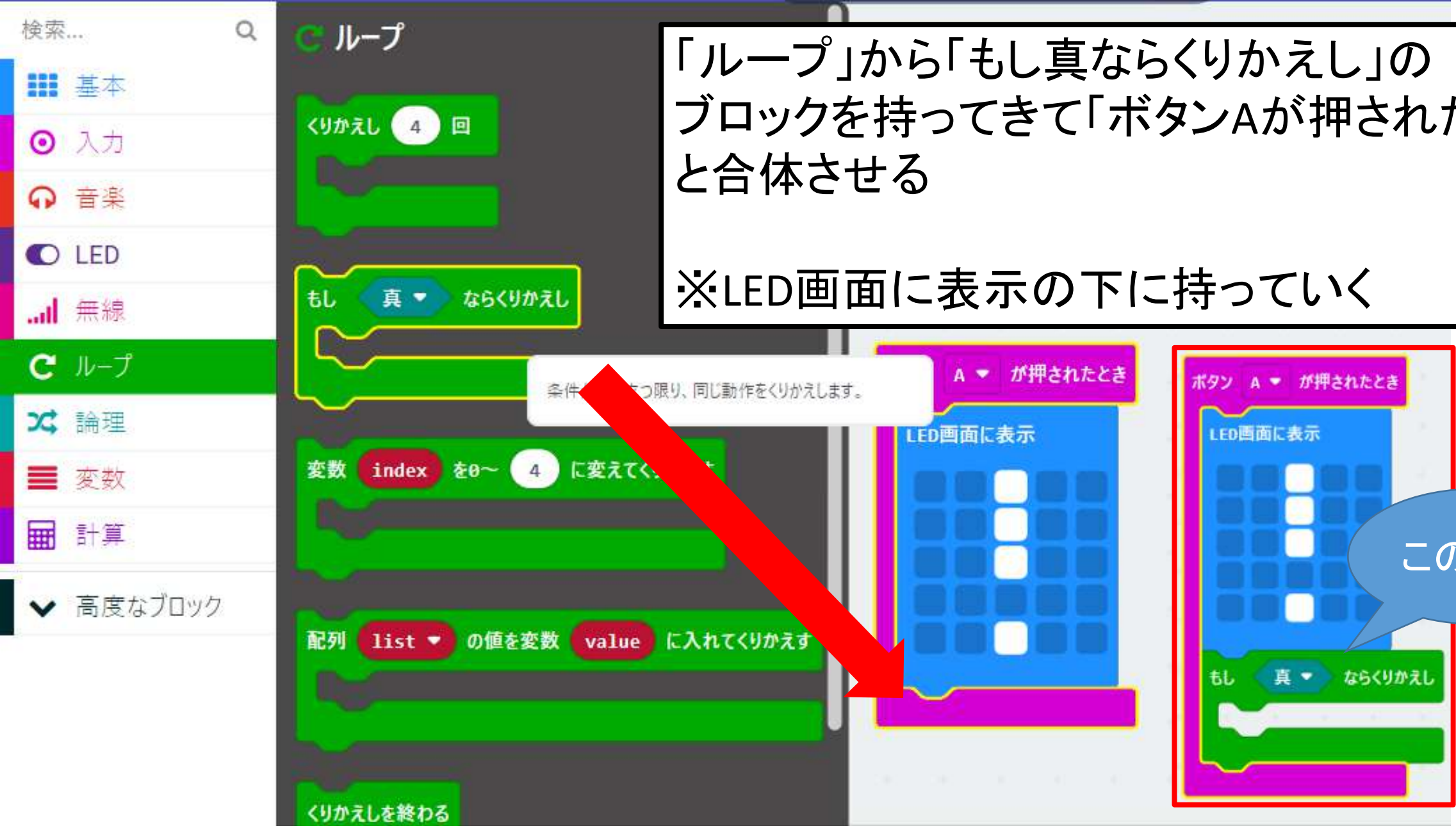


「基本」から「LED画面に表示」の
ブロックを持ってきて「ボタンAが押されたとき」
と合体させる

青いマスをクリックして形を「 ！ 」にする

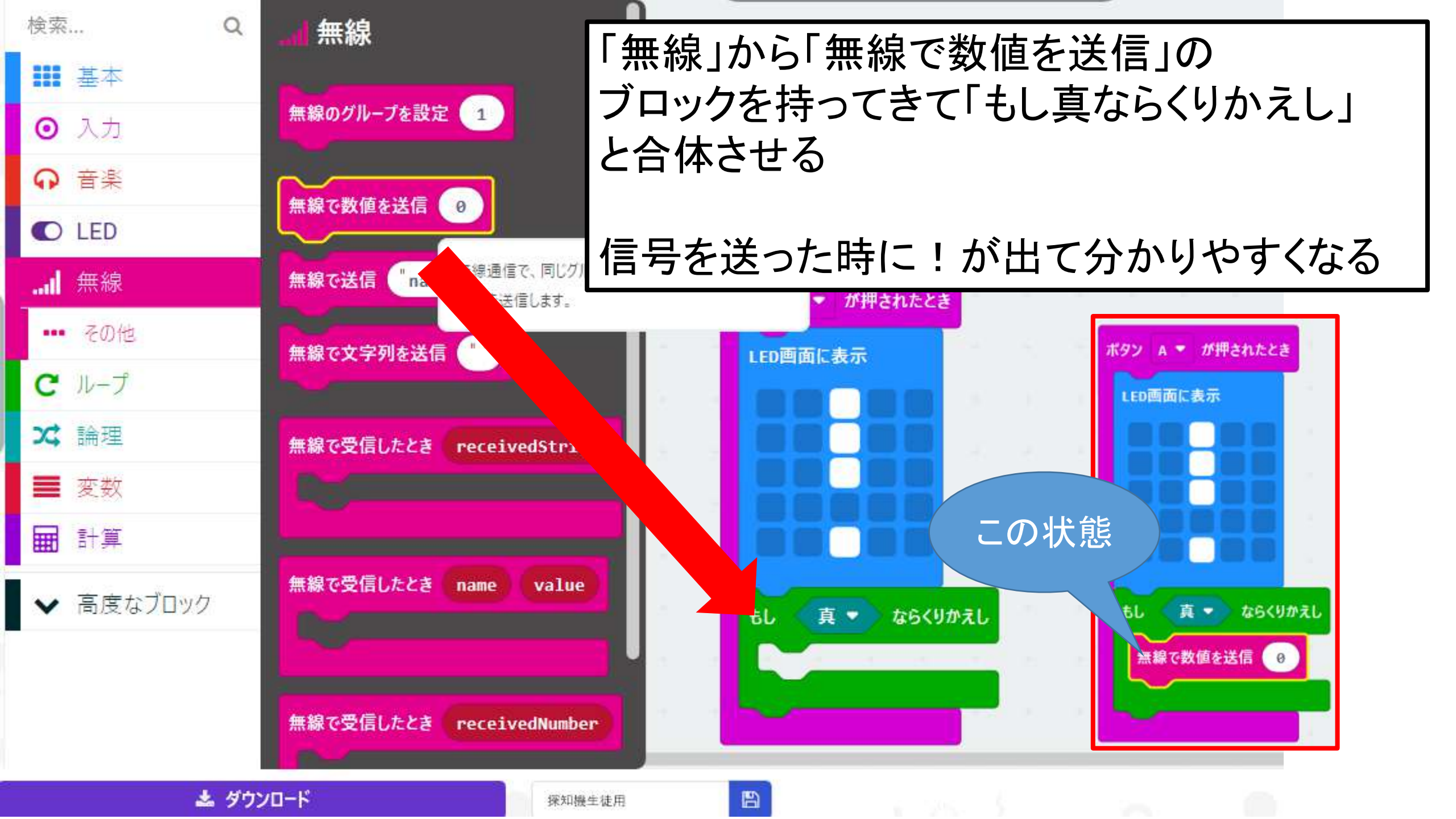
「ループ」から「もし真ならくりかえし」の
ブロックを持ってきて「ボタンAが押されたとき」
と合体させる

※LED画面に表示の下に持っていく



「無線」から「無線で数値を送信」の
ブロックを持ってきて「もし真ならくりかえし」
と合体させる

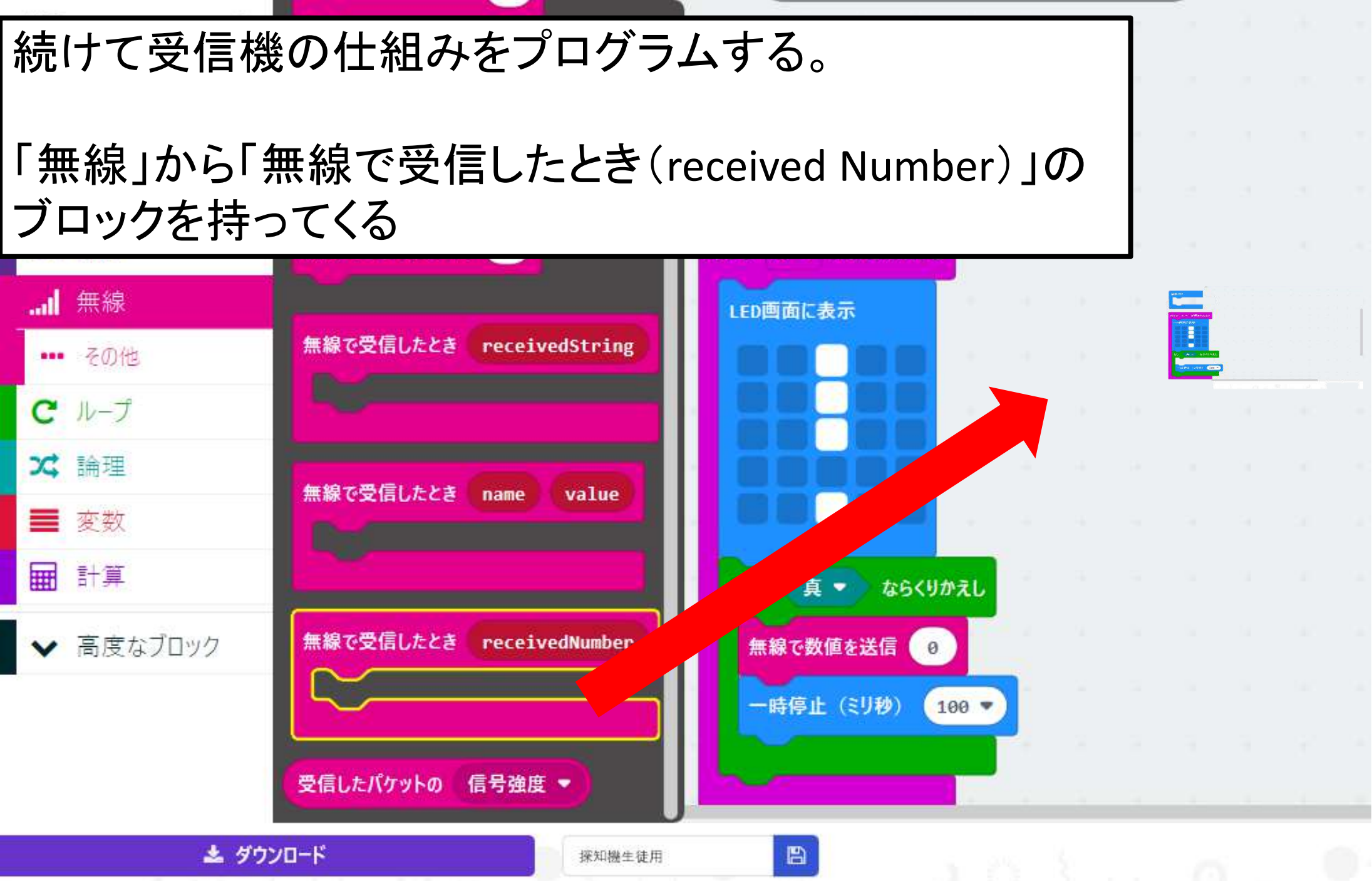
信号を送った時に！が出て分かりやすくなる



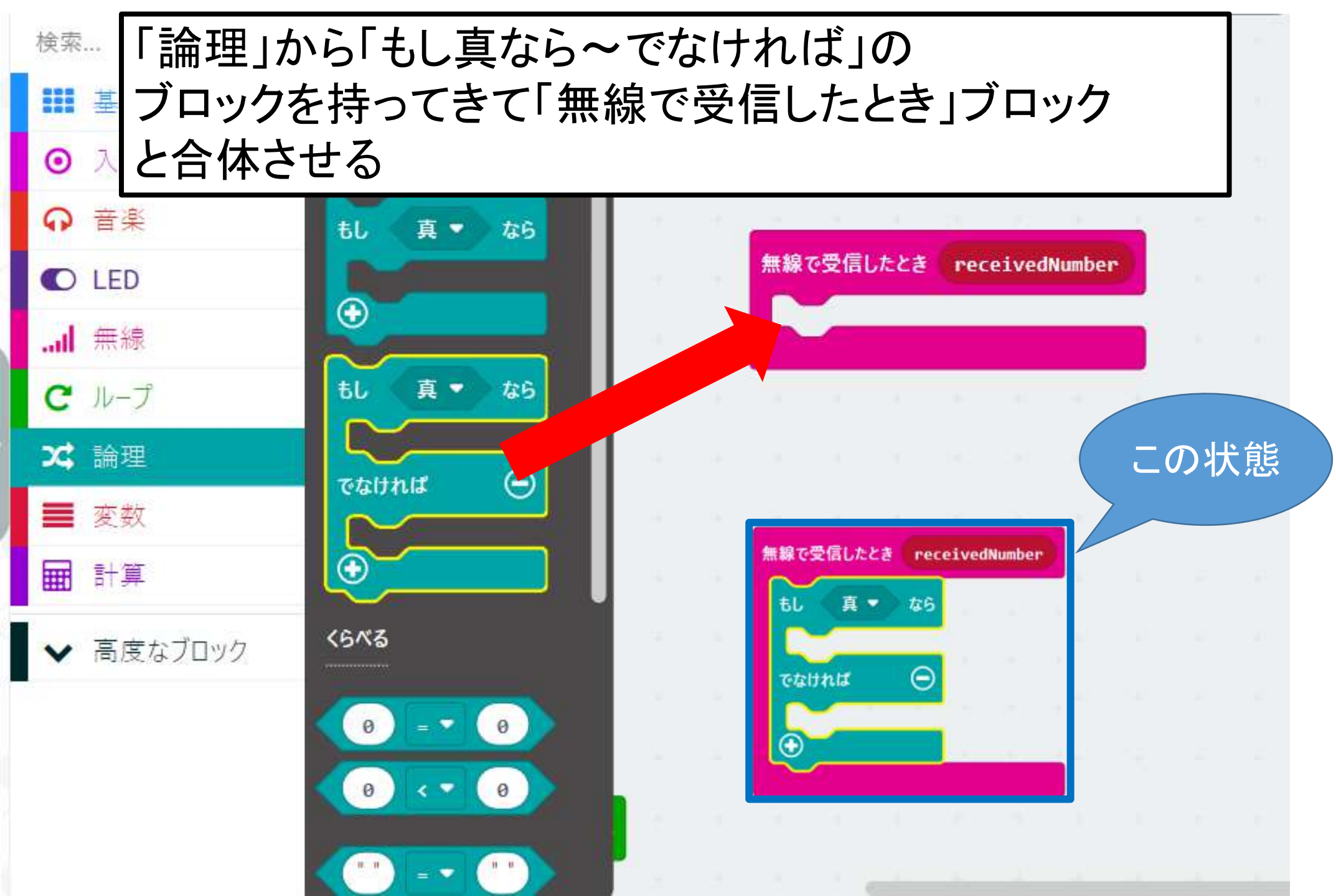

検索...
基本
入力
音楽
LED
無線
ループ
論理
変数
計算
高度なブロック
アイコンを表示
文字列を表示
表示を消す
ずっと
最初だけ
一時停止（ミリ秒） 100
矢印を表示 上向き
無線のグループを設定 1
ずっと
もし 真 ならくりかえし
無線で数値を送信 0
「基本」から「一時停止（ミリ秒）」の
ブロックを持ってきて「もし真ならくりかえし」
と合体させる

続けて受信機の仕組みをプログラムする。

「無線」から「無線で受信したとき（received Number）」のブロックを持ってくる

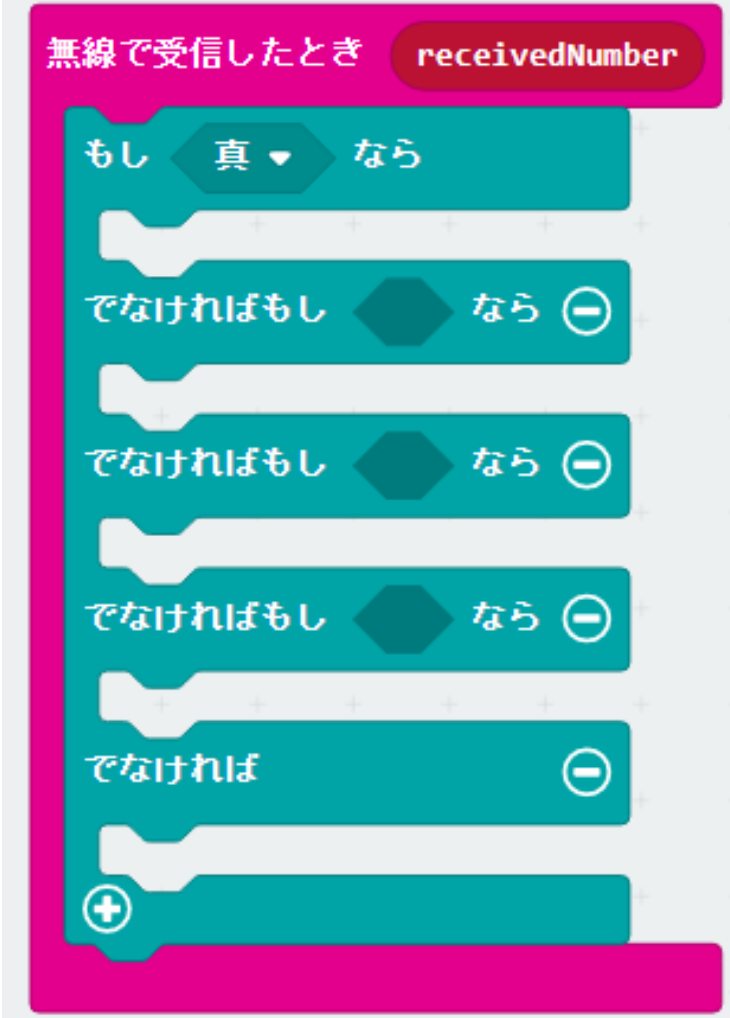


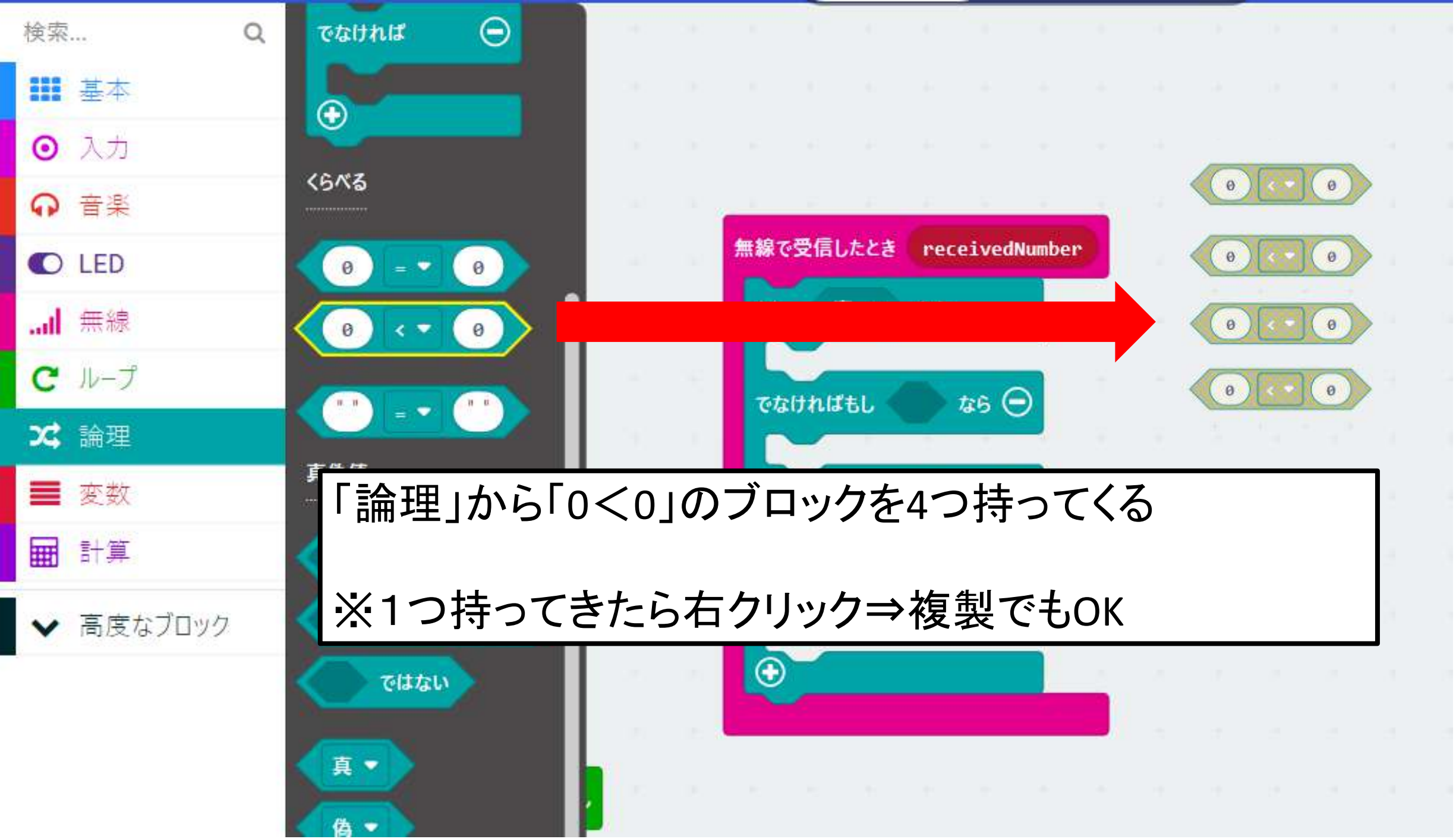
「論理」から「もし真なら～でなければ」の
ブロックを持ってきて「無線で受信したとき」ブロック
と合体させる



「もし真なら～でなければ」ブロックの左下にある
＋ボタンを**3回**押す

ここ

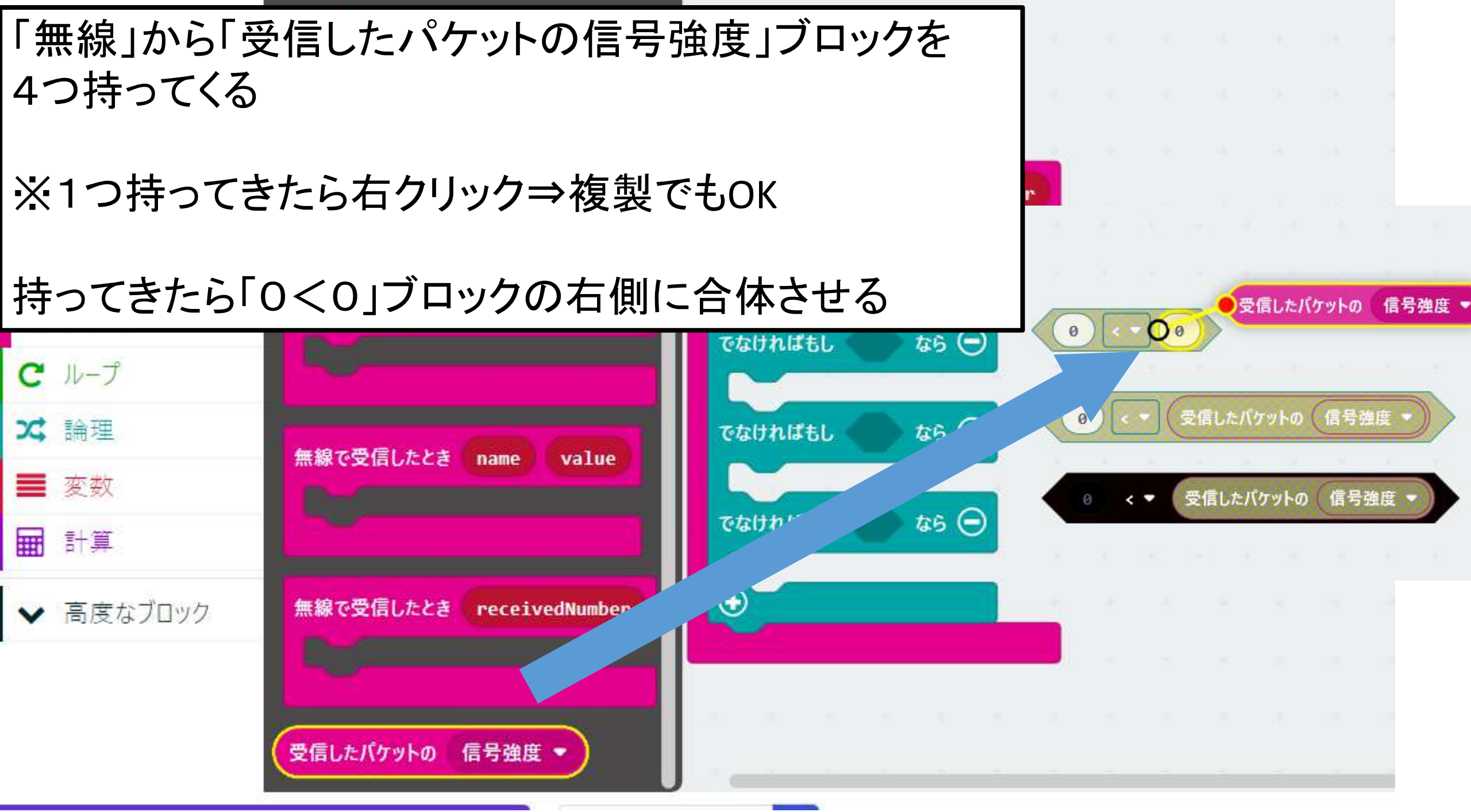


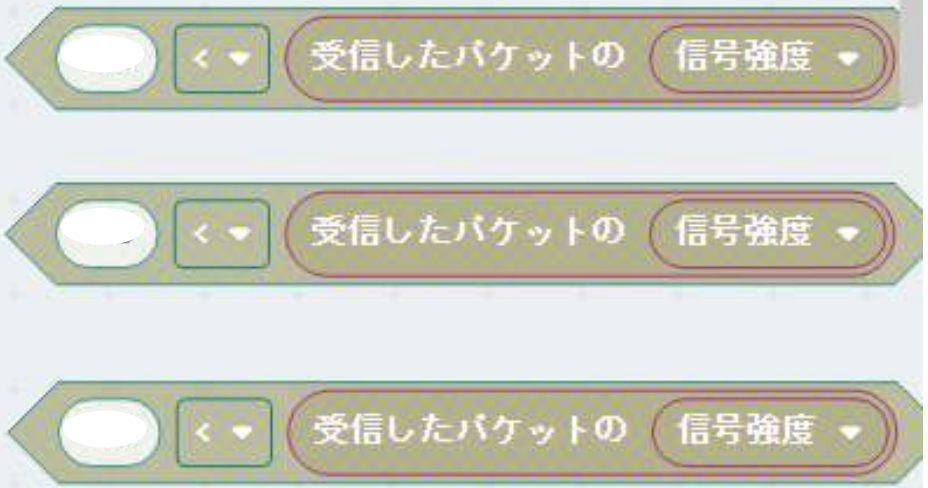

検索...
基本
入力
音楽
LED
無線
ループ
論理
変数
計算
高度なブロック
でなければ
くらべる
0 = 0
0 < 0
" " = " "
ではない
真
偽
無線で受信したとき receivedNumber
でなければもし なら
「論理」から「0＜0」のブロックを4つ持ってくる
※1つ持ってきたら右クリック⇒複製でもOK

「無線」から「受信したパケットの信号強度」ブロックを
4つ持ってくる

※1つ持ってきたら右クリック⇒複製でもOK

持ってきたら「0＜0」ブロックの右側に合体させる



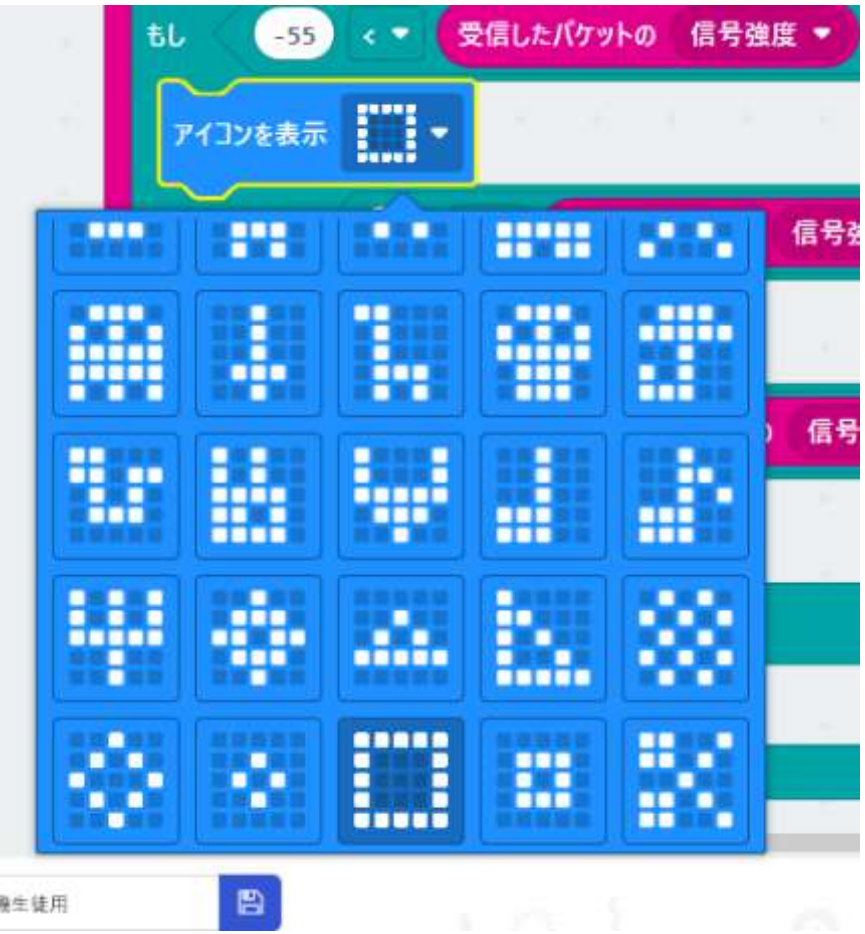


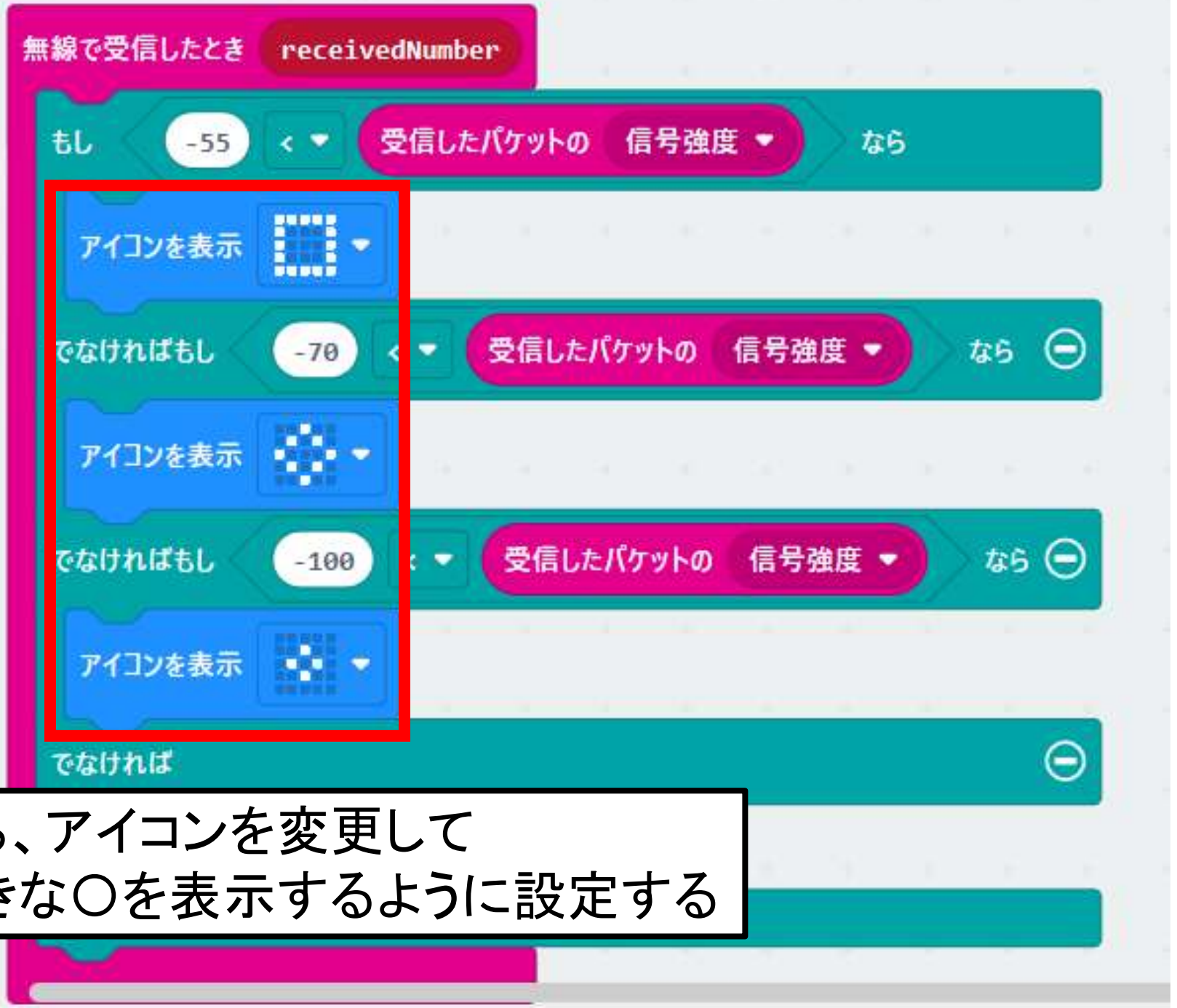
合体した「0＜受信したパケット」ブロックを
「もし～なら」ブロックにそれぞれ合体させる

「0＜受信したパケット」ブロックを
上から-45、-55、-65、-70に書きかえる

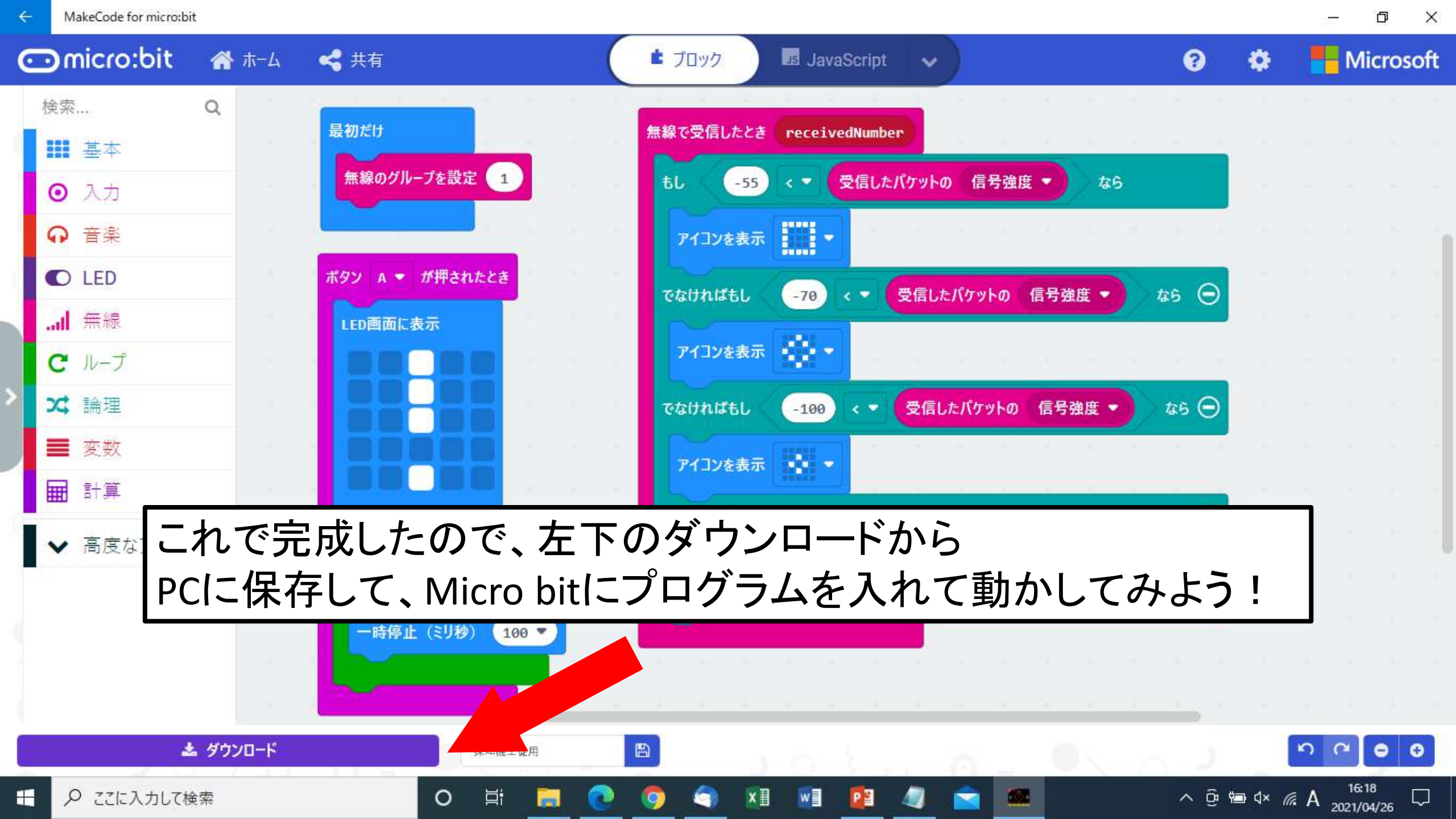

検索...
基本
入力
音楽
LED
無線
ループ
論理
変数
計算
高度なブロック
LED画面に表
アイコンを表示
選択されたアイコンを、LED画面に表示します。
表示を消す
ずっと
でなければもし -70 < 受信したパケットの 信号強度 なら
でなければもし -100 < 受信したパケットの 信号強度 なら
でなければ
「基本」から「アイコンを表示」ブロックを4つ
「もし～なら」ブロックの条件の部分に合体させる
「でなければ」の部分には表示を消すを入れる

Micro bitでは、信号の強さが-128～-42の間で
表されることになっているため、これで送信機に
近づくほどアイコンが表示されるようにすることができた





アイコンを表示の▼から、アイコンを変更して
送信機に近づくほど大きな〇を表示するように設定する


MakeCode for micro:bit
micro:bit
ホーム
共有
ブロック
JavaScript
Microsoft
検索...
基本
入力
音楽
LED
無線
ループ
論理
変数
計算
最初だけ
無線のグループを設定 1
ボタン A が押されたとき
LED画面に表示
一時停止（ミリ秒） 100
無線で受信したとき receivedNumber
もし -55 < 受信したパケットの 信号強度 なら
アイコンを表示
でなければもし -70 < 受信したパケットの 信号強度 なら
アイコンを表示
でなければもし -100 < 受信したパケットの 信号強度 なら
アイコンを表示
これで完成したので、左下のダウンロードから
PCに保存して、Micro bitにプログラムを入れて動かしてみよう！
ダウンロード
ここに入力して検索
16:18
2021/04/26

ドライブゲームを作ろう

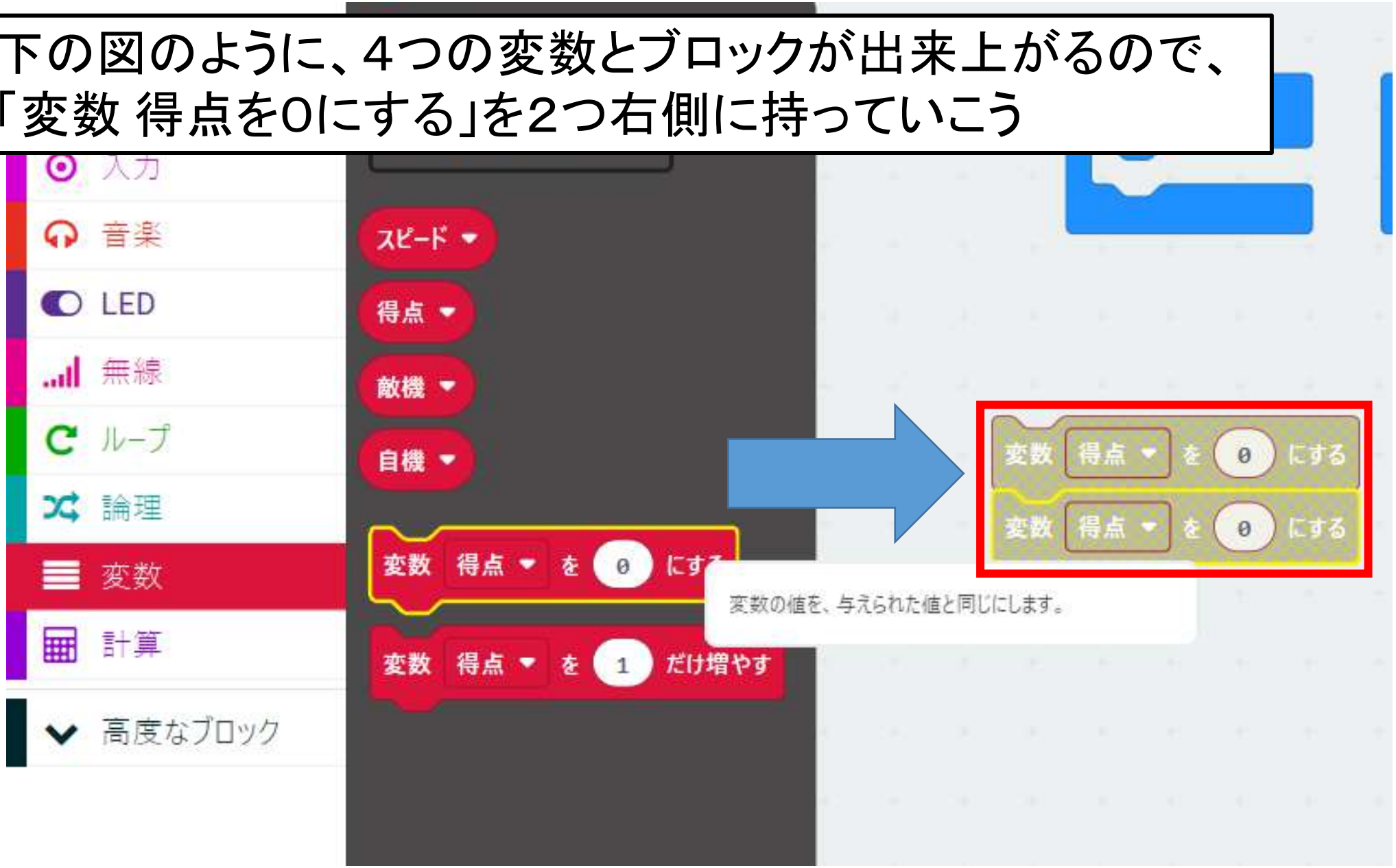
LED機能を使って、上から迫ってくる車を避けるゲームを作ろう

まずは、プログラムの入れ物になる「変数」を作成する

左側のブロック一覧から「変数」をクリックして、「変数を追加する...」をクリックしよう

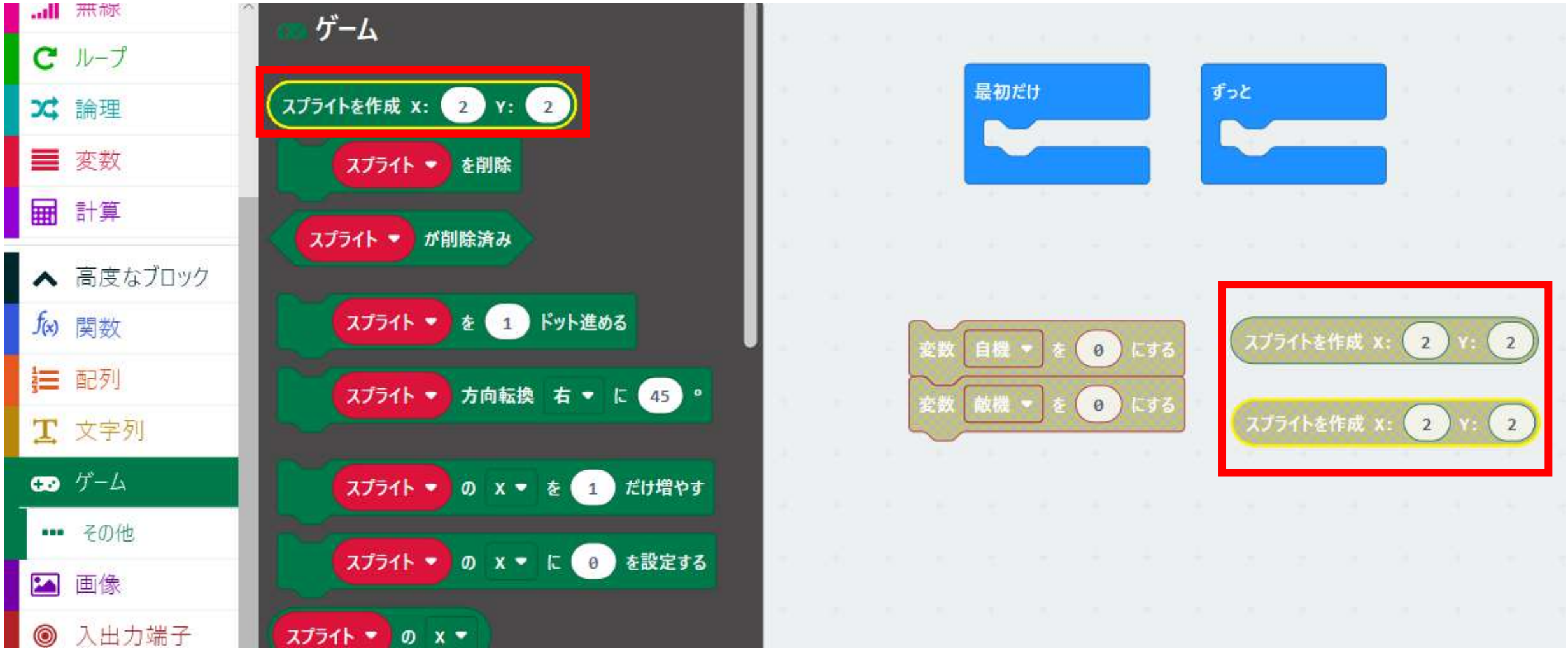
「自機」「敵機」「スピード」「得点」の4つの変数を作成しよう

下の図のように、4つの変数とブロックが出来上がるので、
「変数 得点を0にする」を2つ右側に持っていこう

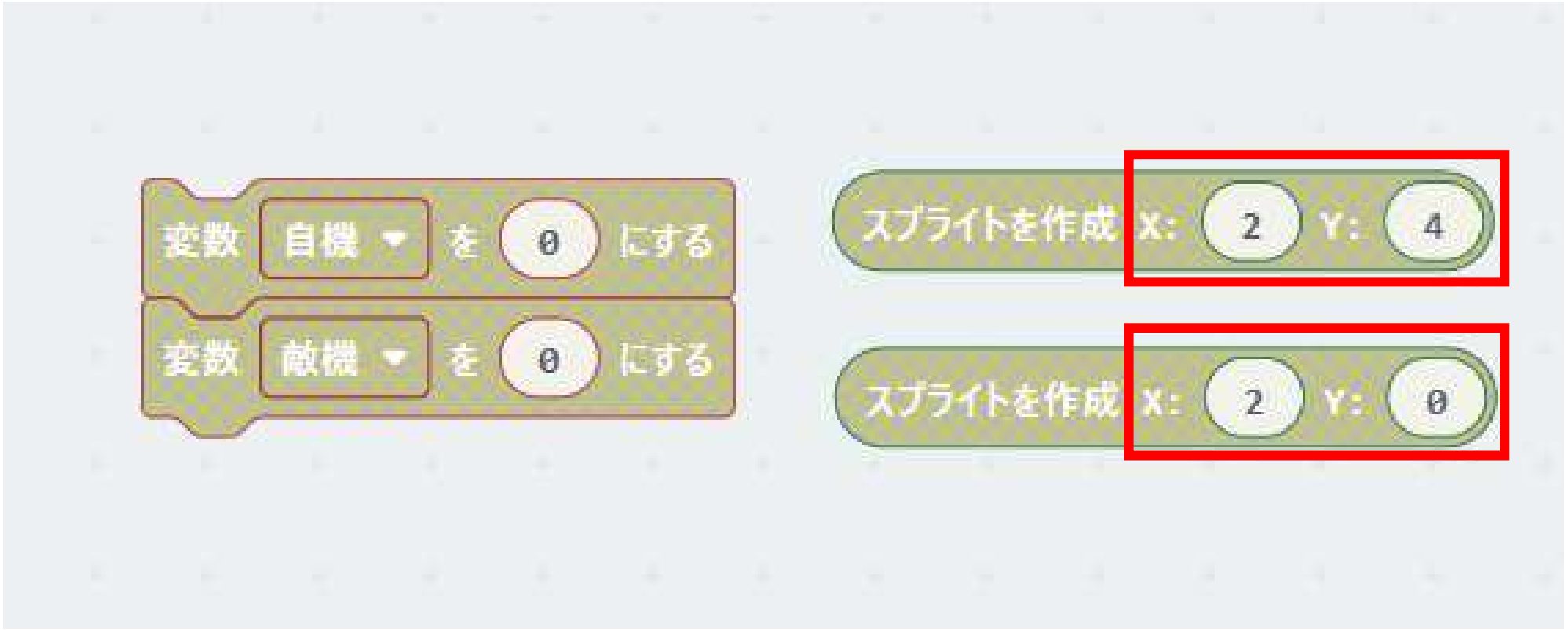


持ってきた変数のブロックを、それぞれ下の図のように変更しよう

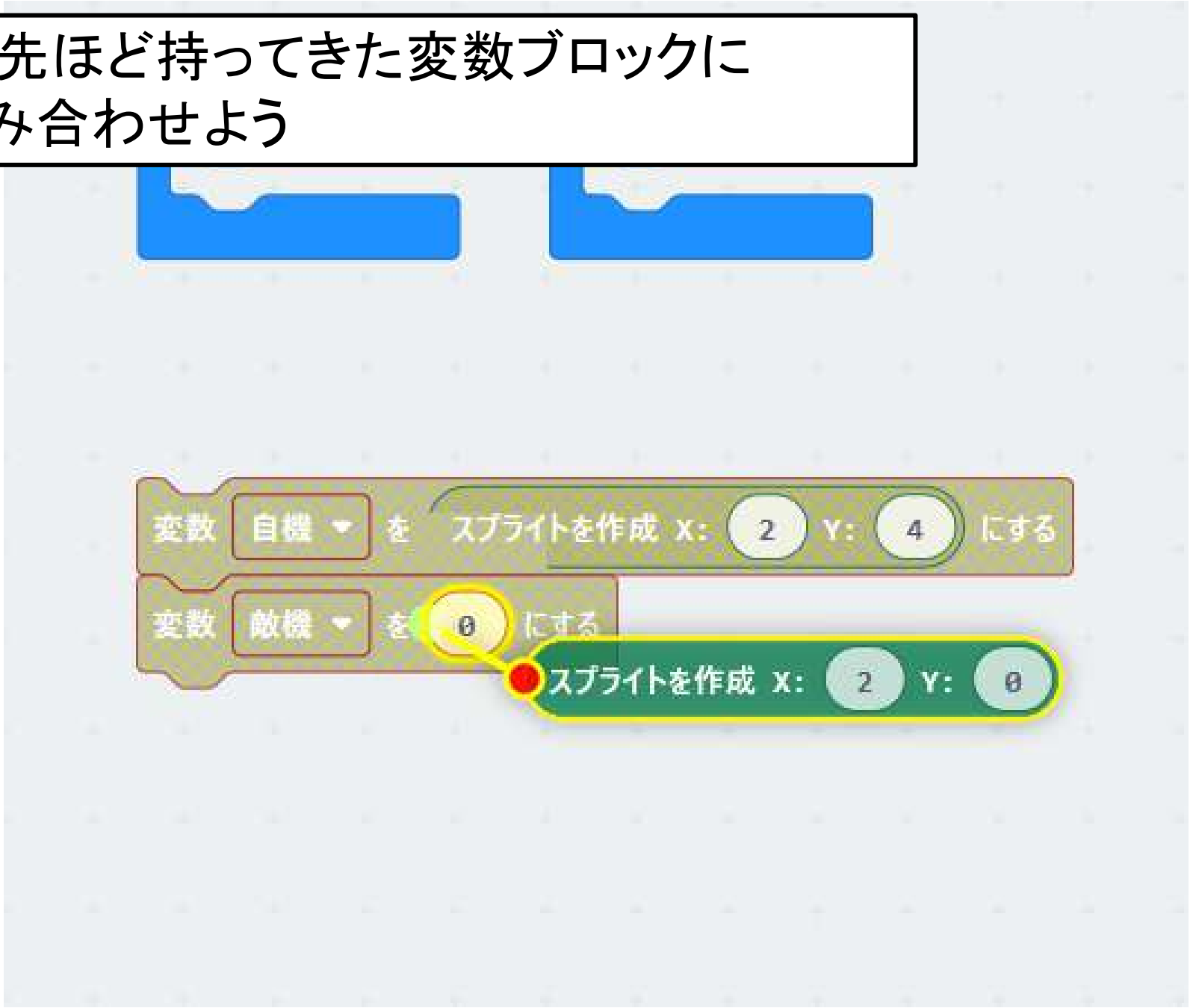
「ゲーム」から、「スプライトを作成 X:〇 Y:〇」を
2つ右側に持っていこう



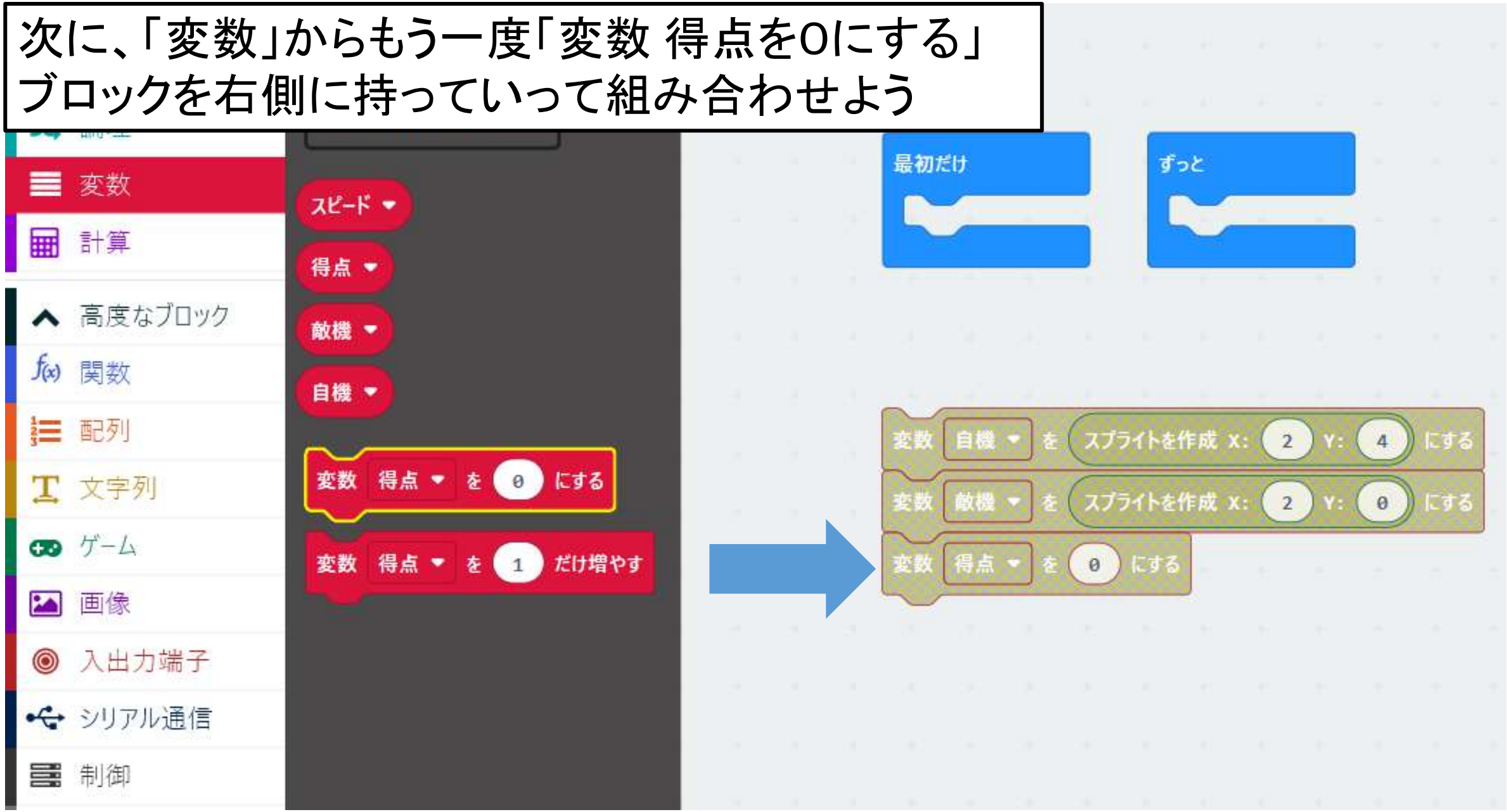
持ってきたブロックを、
「スプライトを作成 X:2　Y:4」と
「スプライトを作成 X:2　Y:0」に変更しよう

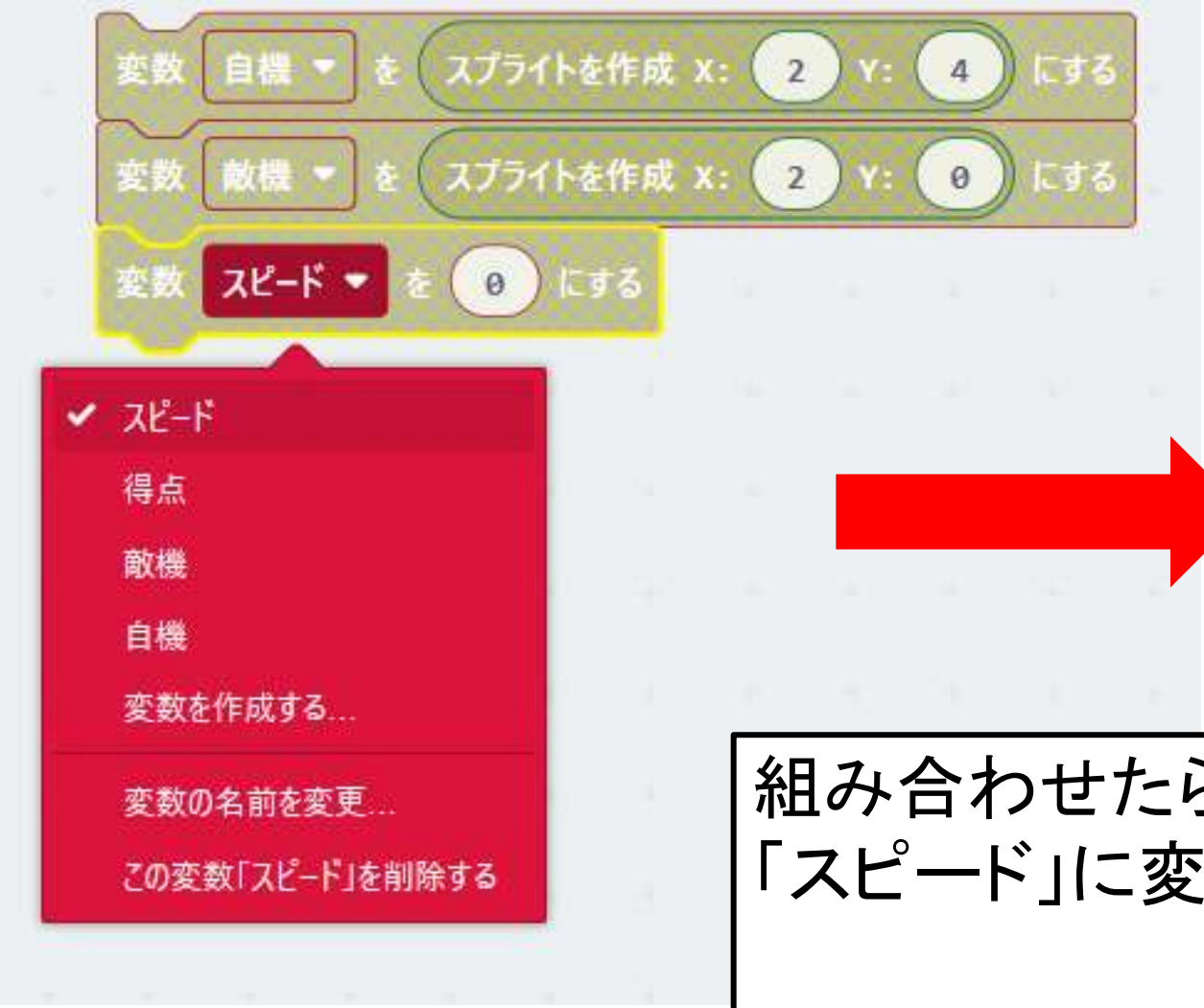


変更したら、先ほど持ってきた変数ブロックに
それぞれ組み合わせよう



次に、「変数」からもう一度「変数 得点を0にする」
ブロックを右側に持っていって組み合わせよう



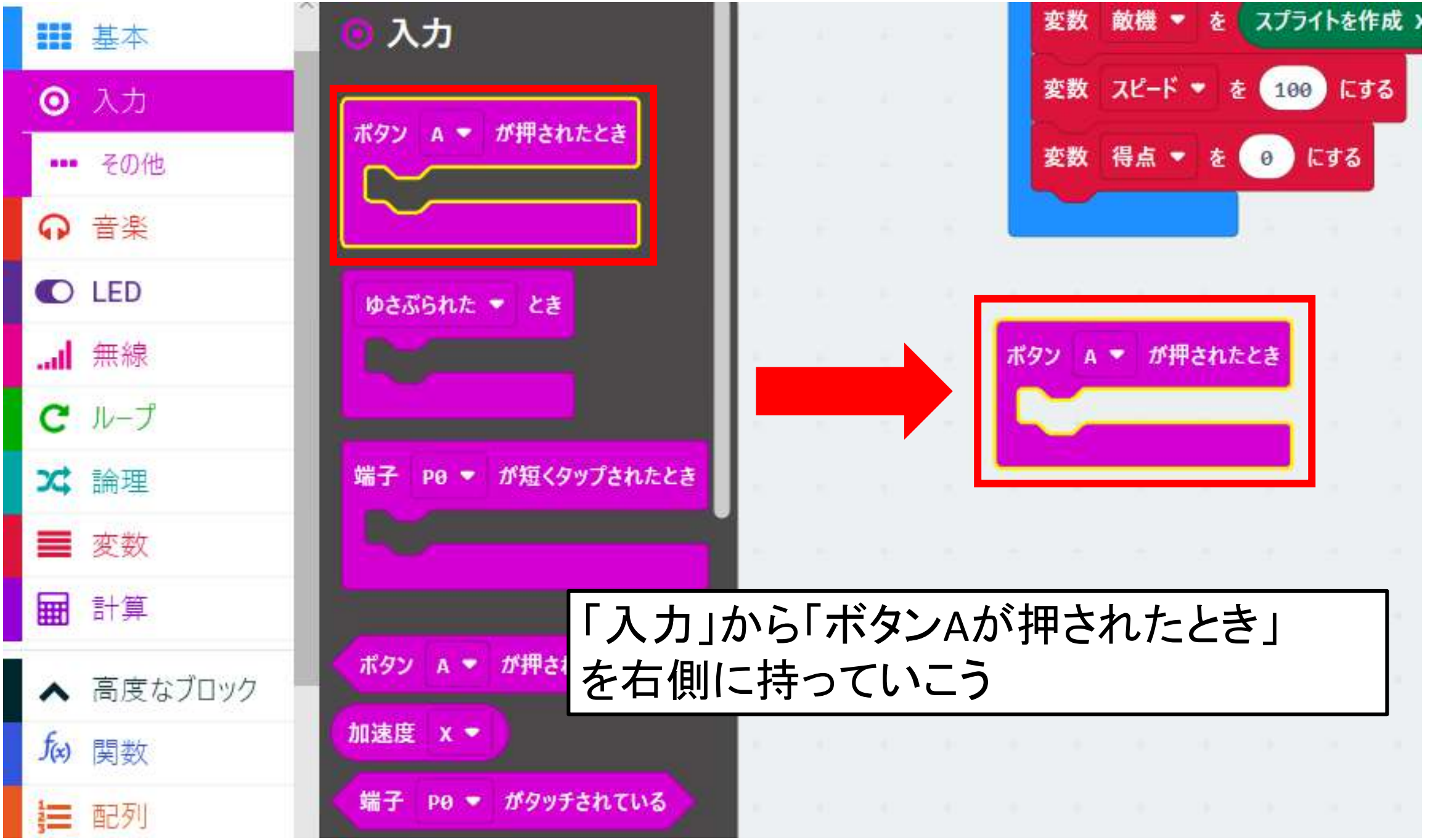


組み合わせたら、「得点」の部分をクリックして
「スピード」に変更しよう。

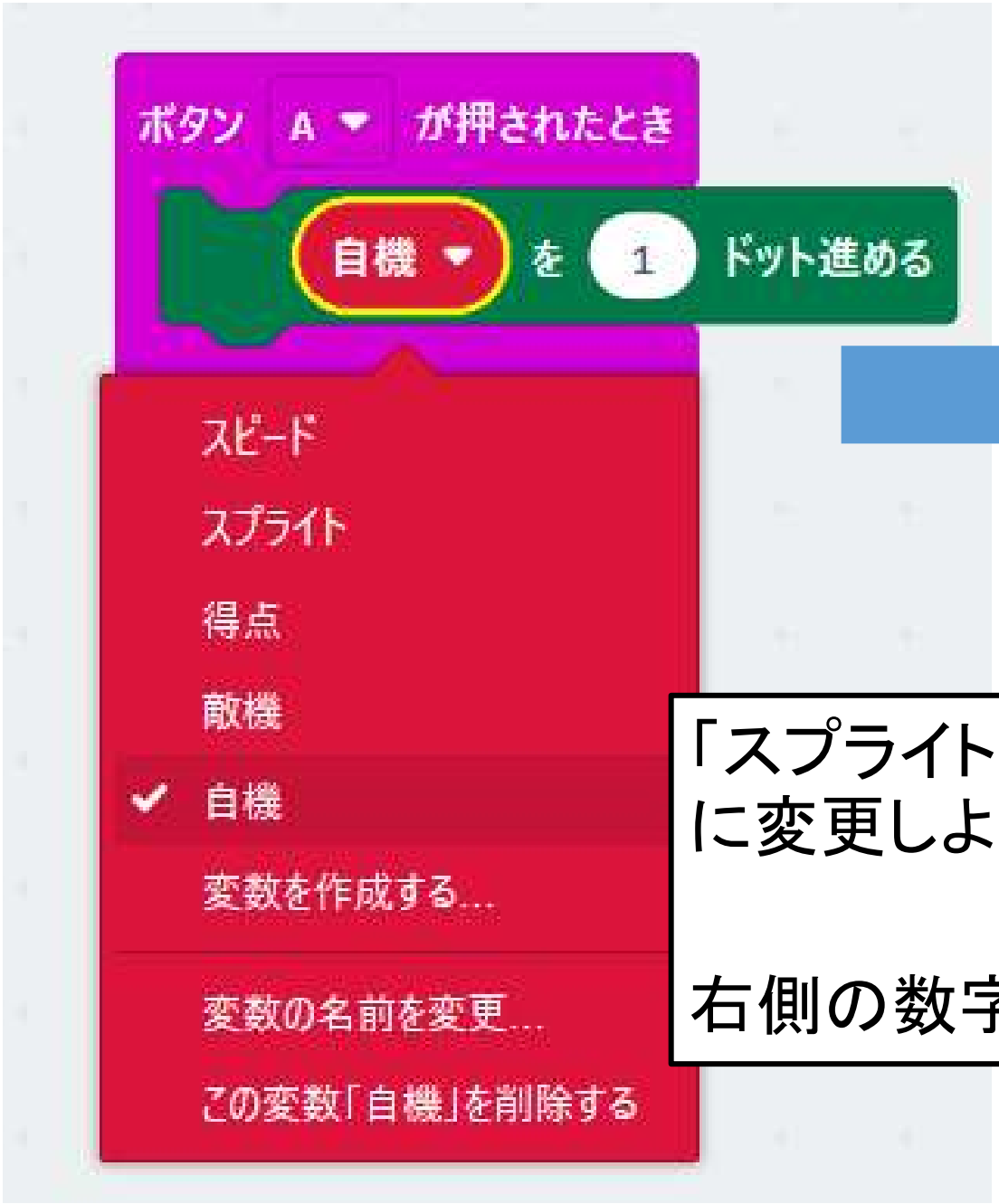
続けて、0と数字が入っているところを100に
変更しよう

さらに「変数」から「変数 得点を0にする」
ブロックを右側に持っていって組み合わせよう

組み合わせた変数ブロックを、「最初だけ」ブロックと合体させよう


基本
入力
その他
音楽
LED
無線
ループ
論理
変数
計算
高度なブロック
関数
配列
入力
ボタン A が押されたとき
ゆさぶられた とき
端子 P0 が短くタップされたとき
ボタン A が押さ
加速度 X
端子 P0 がタッチされている
変数 敵機 を スプライトを作成
変数 スピード を 100 にする
変数 得点 を 0 にする
ボタン A が押されたとき
「入力」から「ボタンAが押されたとき」
を右側に持っていこう

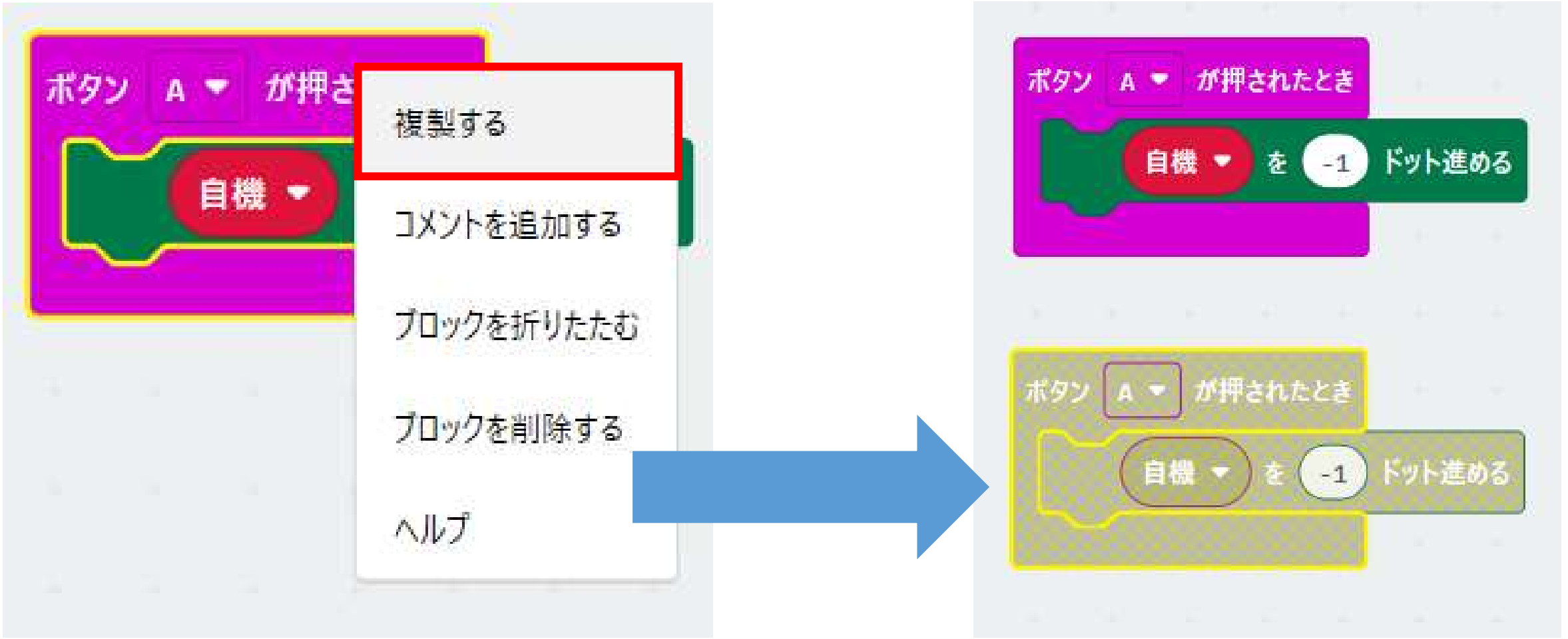

ループ
論理
変数
計算
高度なブロック
関数
配列
文字列
ゲーム
その他
画像
入出力端子
シリアル通信
制御
拡張機能
ゲーム
スプライトを作成 X: 2 Y: 2
スプライト を削除
スプライト が削除済み
スプライト を 1 ドット進める
スプライト 方向転換 右 に 45 °
スプライト の X を 1 だけ増やす
スプライト の X
スプライト の X
スプライト が他のスプライト にさわっている
変数 敵機 を スプライトを作成 X: 2 Y:
変数 スピード を 100 にする
変数 得点 を 0 にする
ボタン A が押されたとき
スプライト を 1 ドット進める
「ゲーム」から「スプライトを1ドット進める」
を右側に持っていって組み合わせよう



「スプライト」をクリックして「自機を1ドット進める」に変更しよう

右側の数字を1から-1に変更しよう

「ボタンAが押されたとき」を右クリックして
「複製する」をクリックしよう



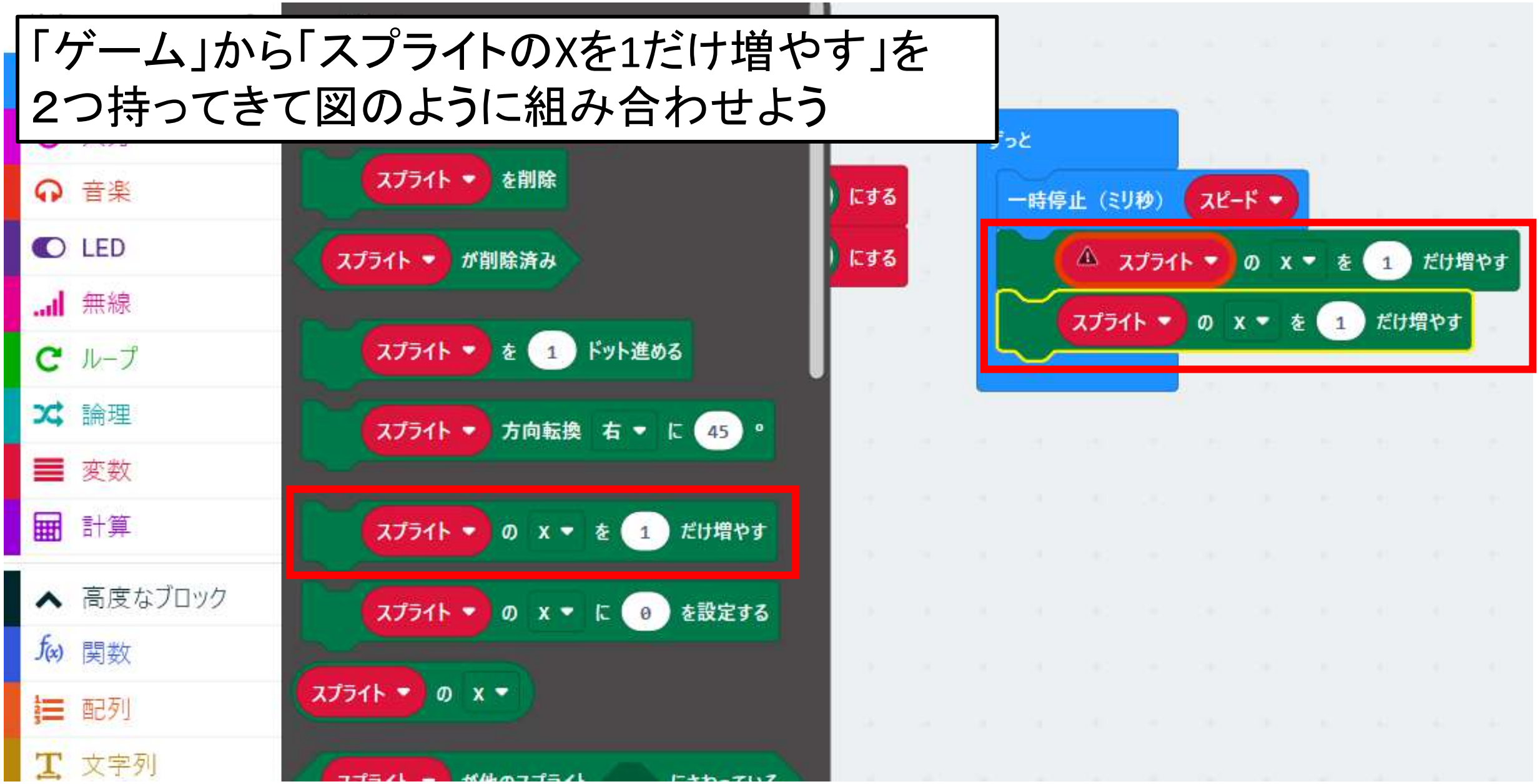
コピーされた方のブロックを
「ボタンBが押されたとき」に変更しよう

右側の数字を-1から1に変更しよう


検索...
基本
入力
音楽
LED
無線
ループ
論理
変数
計算
高度なブロック
関数
配列
文字列
アイコンを表示
文字列を表示 "Hello!"
表示を消す
ずっと
最初だけ
一時停止（ミリ秒） 100
矢印を表示 上向き
成 X: 2 Y: 4 にする
成 X: 2 Y: 0 にする
ずっと
一時停止（ミリ秒） 100
「基本」から「一時停止（ミリ秒）」を
持ってきて「ずっと」ブロックに組み合わせよう

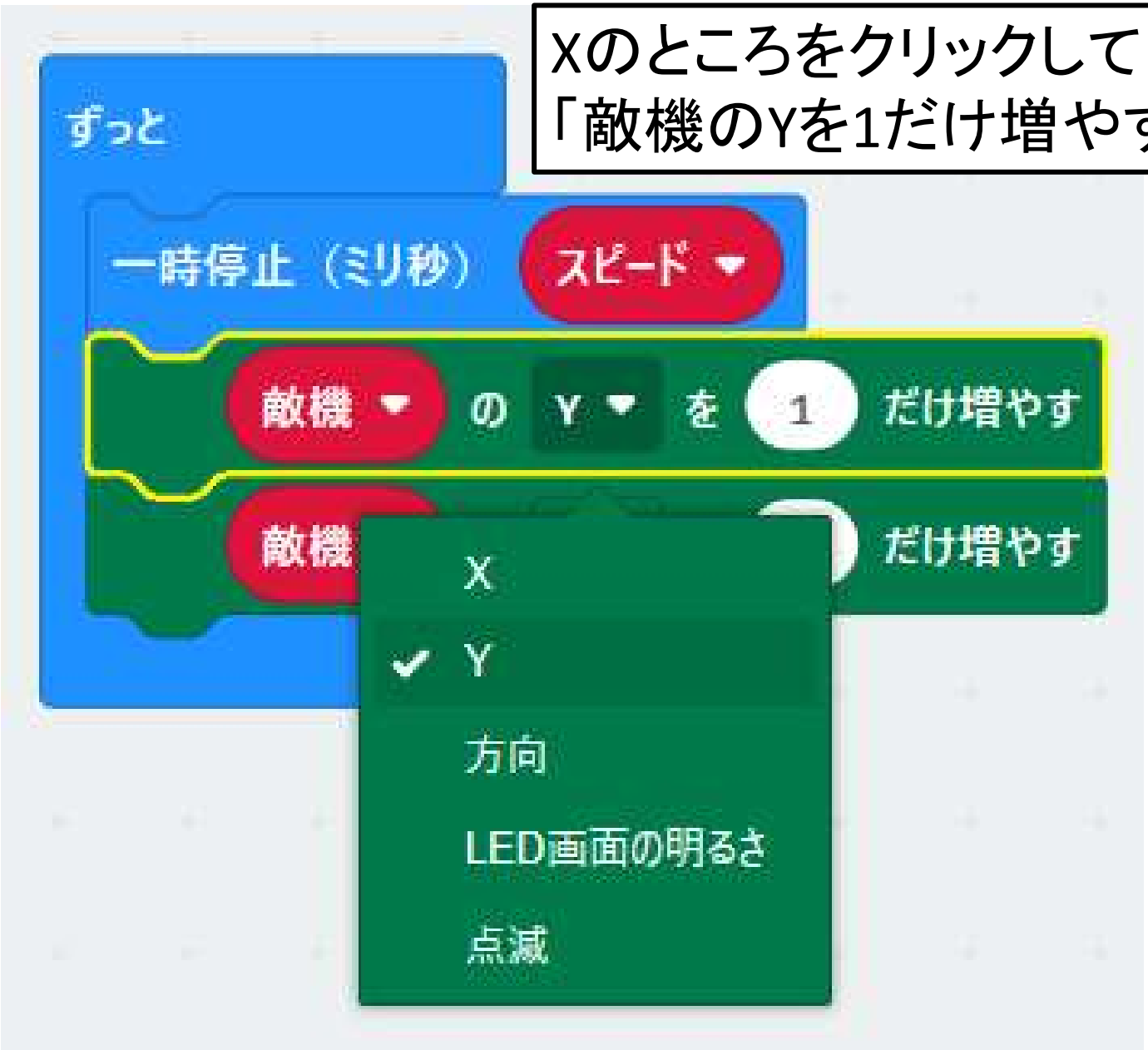

検索...
基本
入力
音楽
LED
無線
ループ
論理
変数
計算
高度なブロック
関数
配列
変数
変数を追加する...
スピード
スプライト
得点
敵機
自機
にする
にする
変数 スプライト
変数 スプライト を 1 だけ増やす
ずっと
一時停止（ミリ秒）
スピード
「変数」から「スピード」を
持ってきて「一時停止」ブロックに組み合わせよう

「ゲーム」から「スプライトのXを1だけ増やす」を
2つ持ってきて図のように組み合わせよう

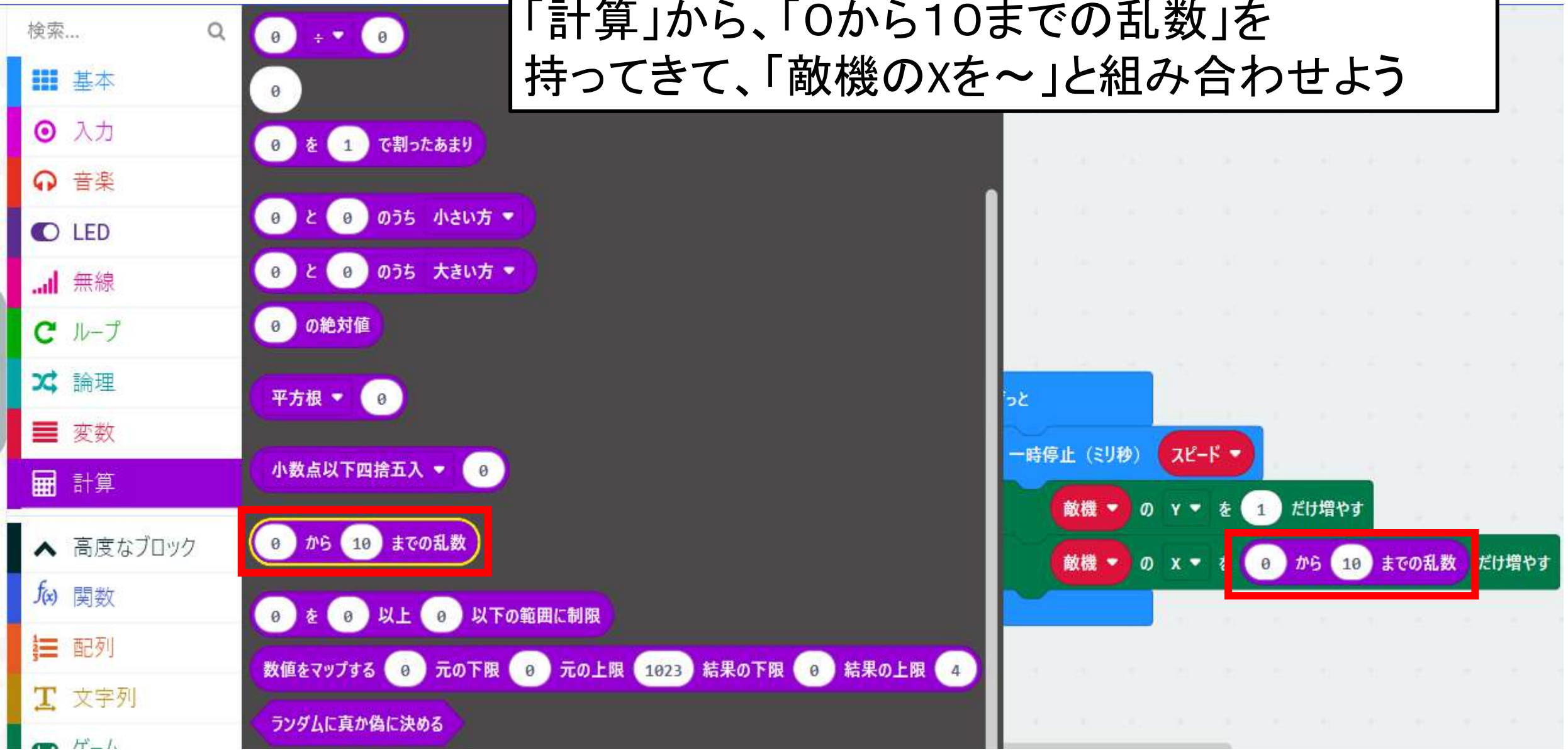


「スプライト」をクリックして、
「敵機のXを1だけ増やす」にどちらも変更しよう

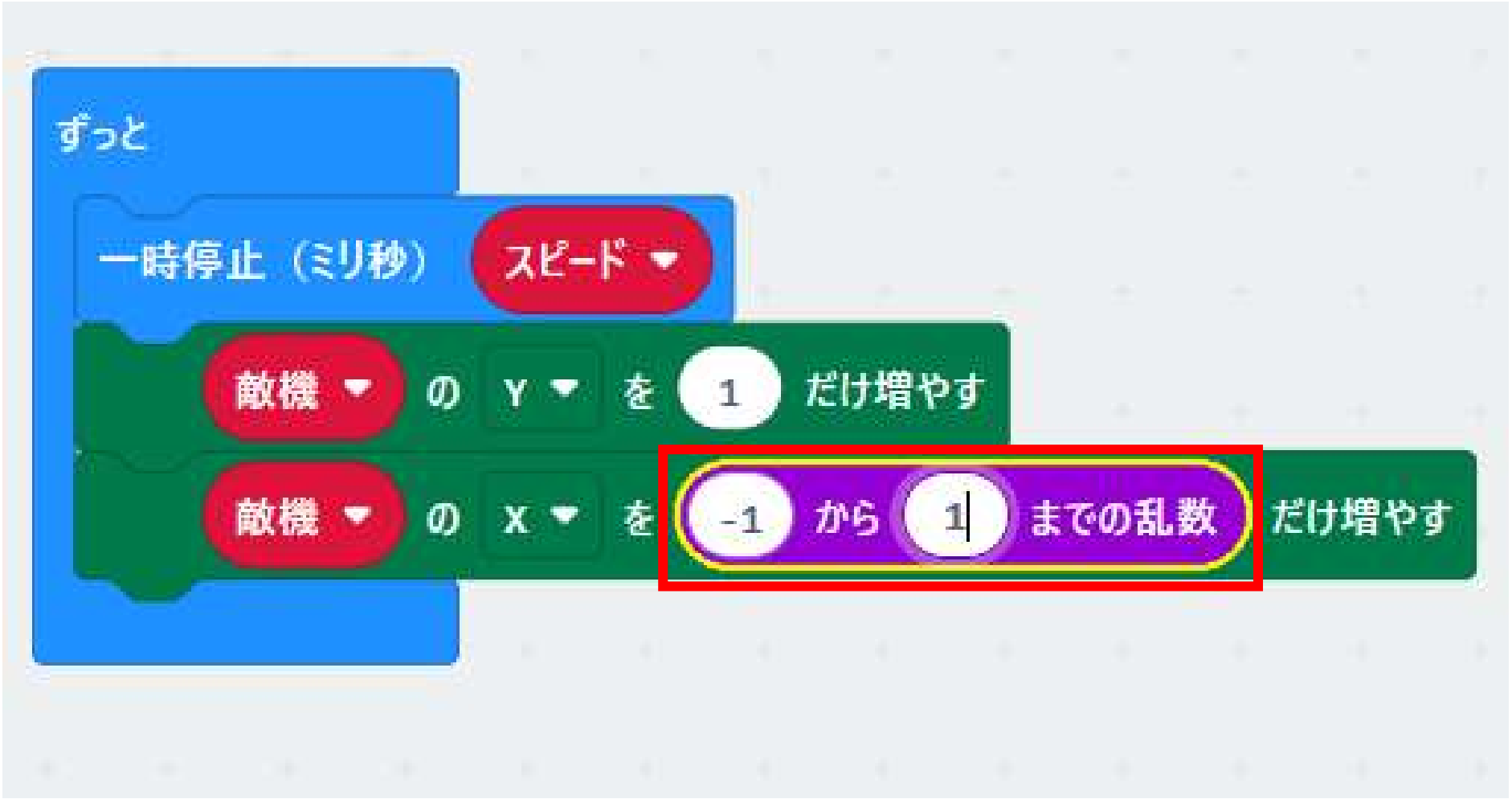
Xのところをクリックして、
「敵機のYを1だけ増やす」に変更しよう

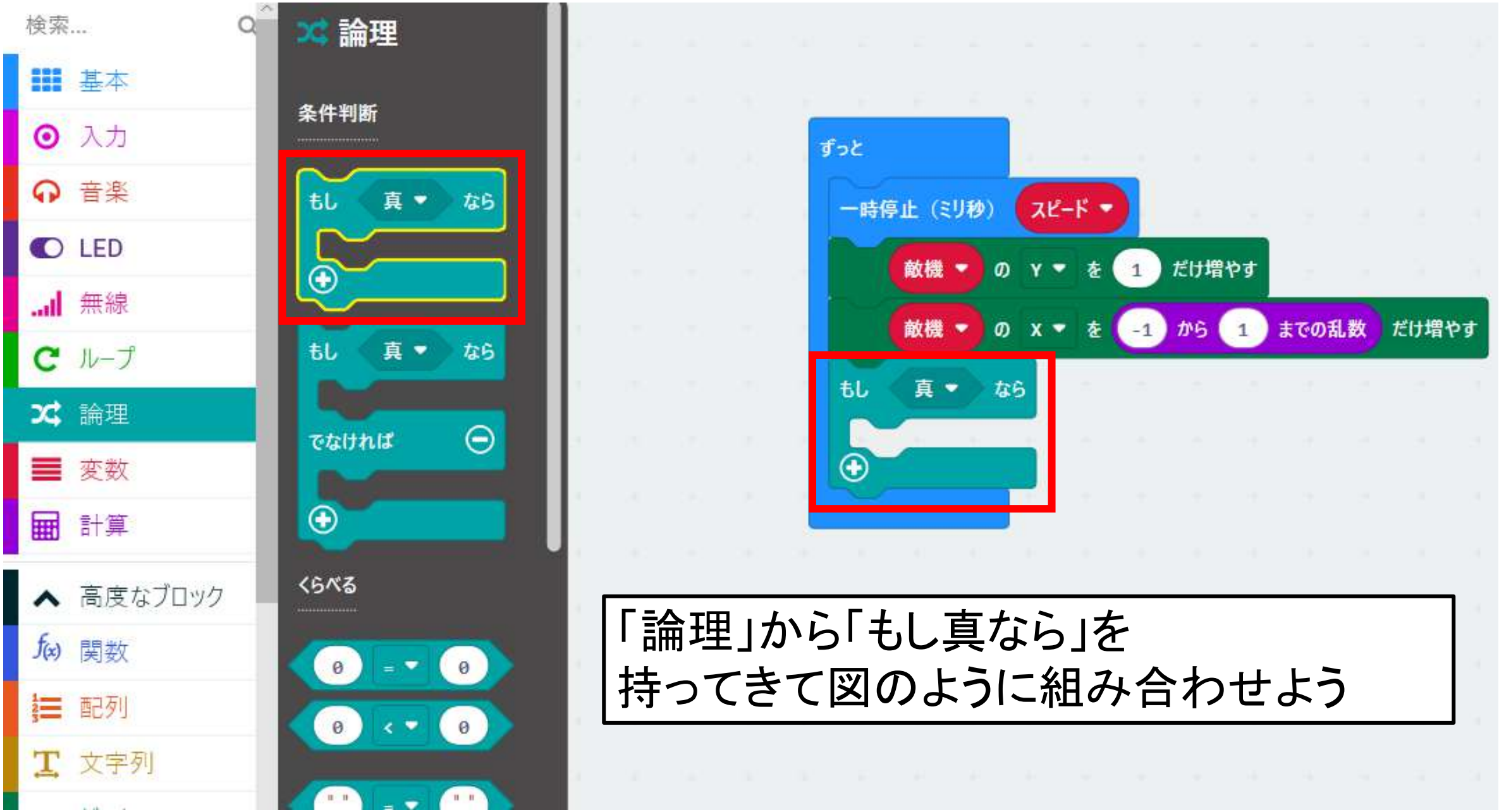


「計算」から、「0から10までの乱数」を
持ってきて、「敵機のXを～」と組み合わせよう

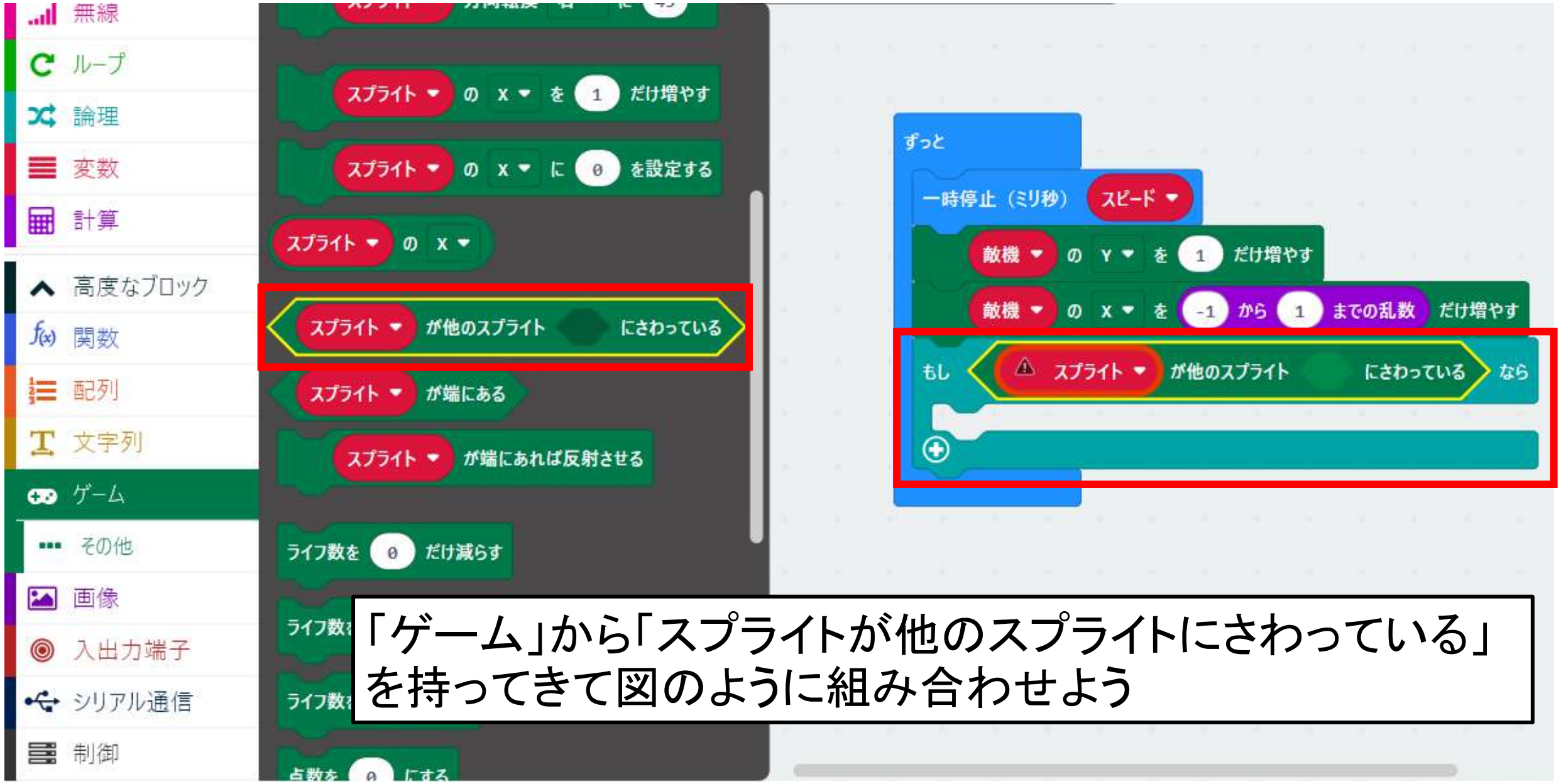


乱数を「-1から1までの乱数」に変更しよう



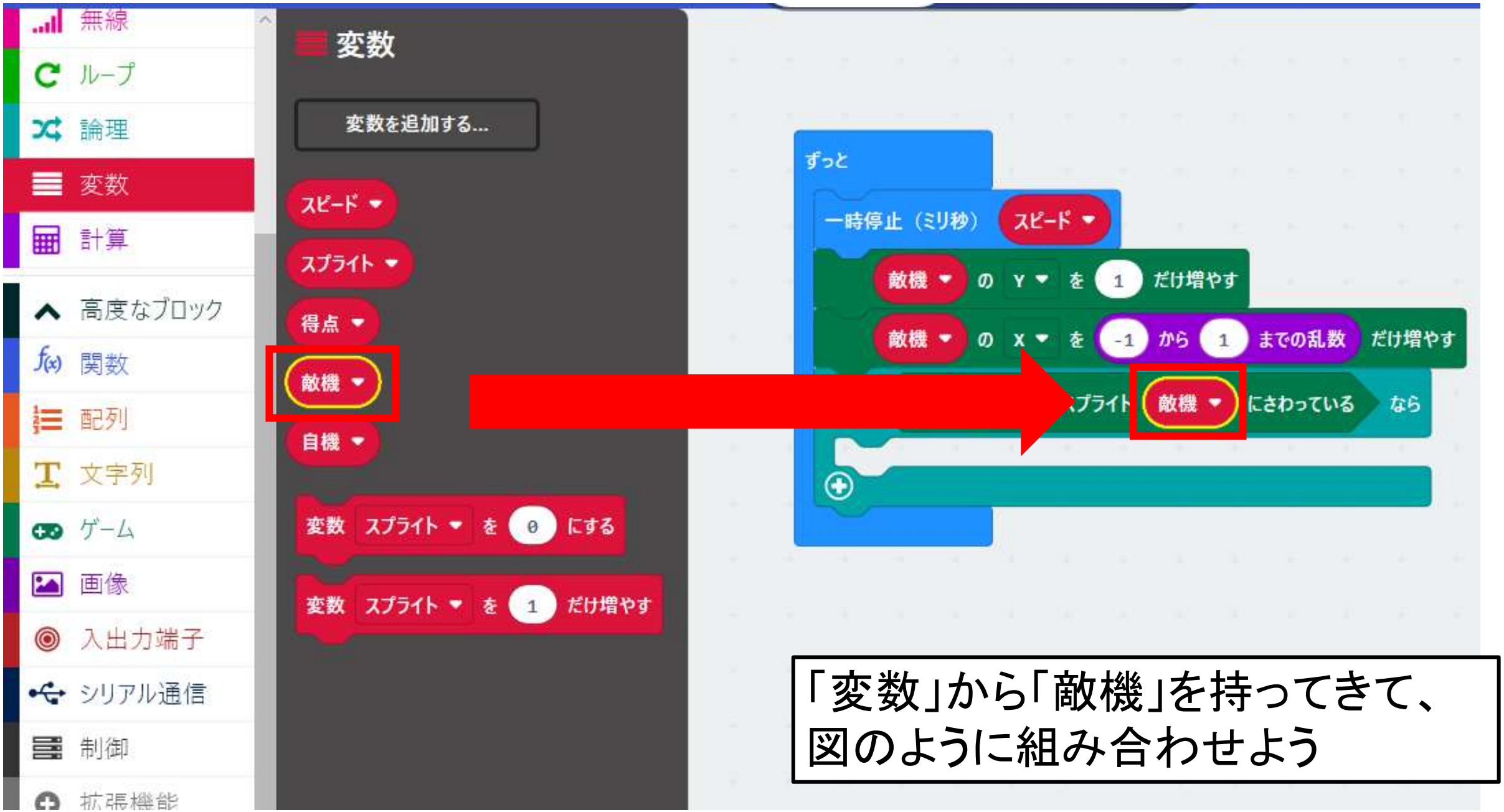


「論理」から「もし真なら」を
持ってきて図のように組み合わせよう



「ゲーム」から「スプライトが他のスプライトにさわっている」を持ってきて図のように組み合わせよう

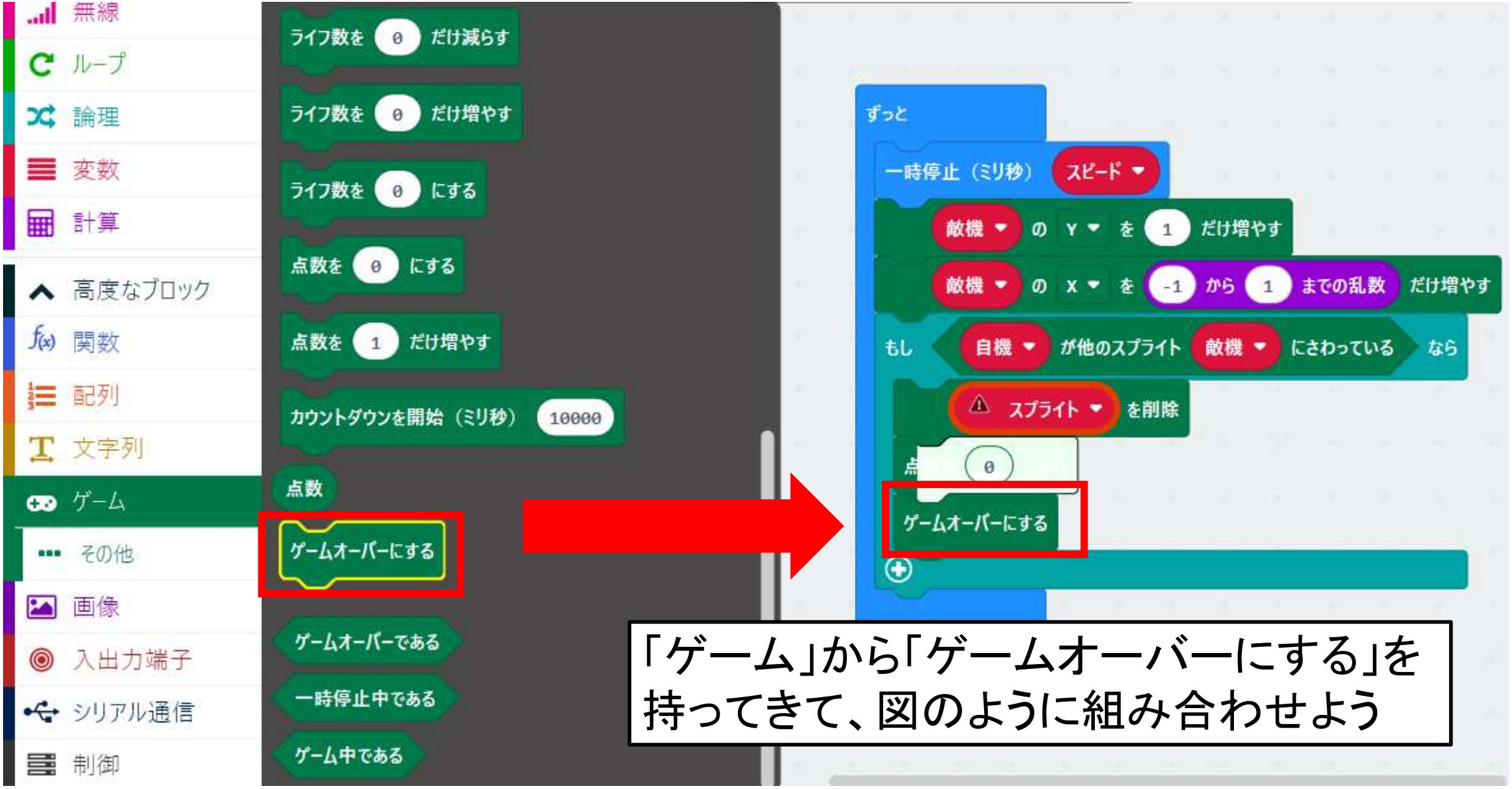
「スプライト」をクリックして、
「自機が他のスプライトにさわっている」に変更しよう



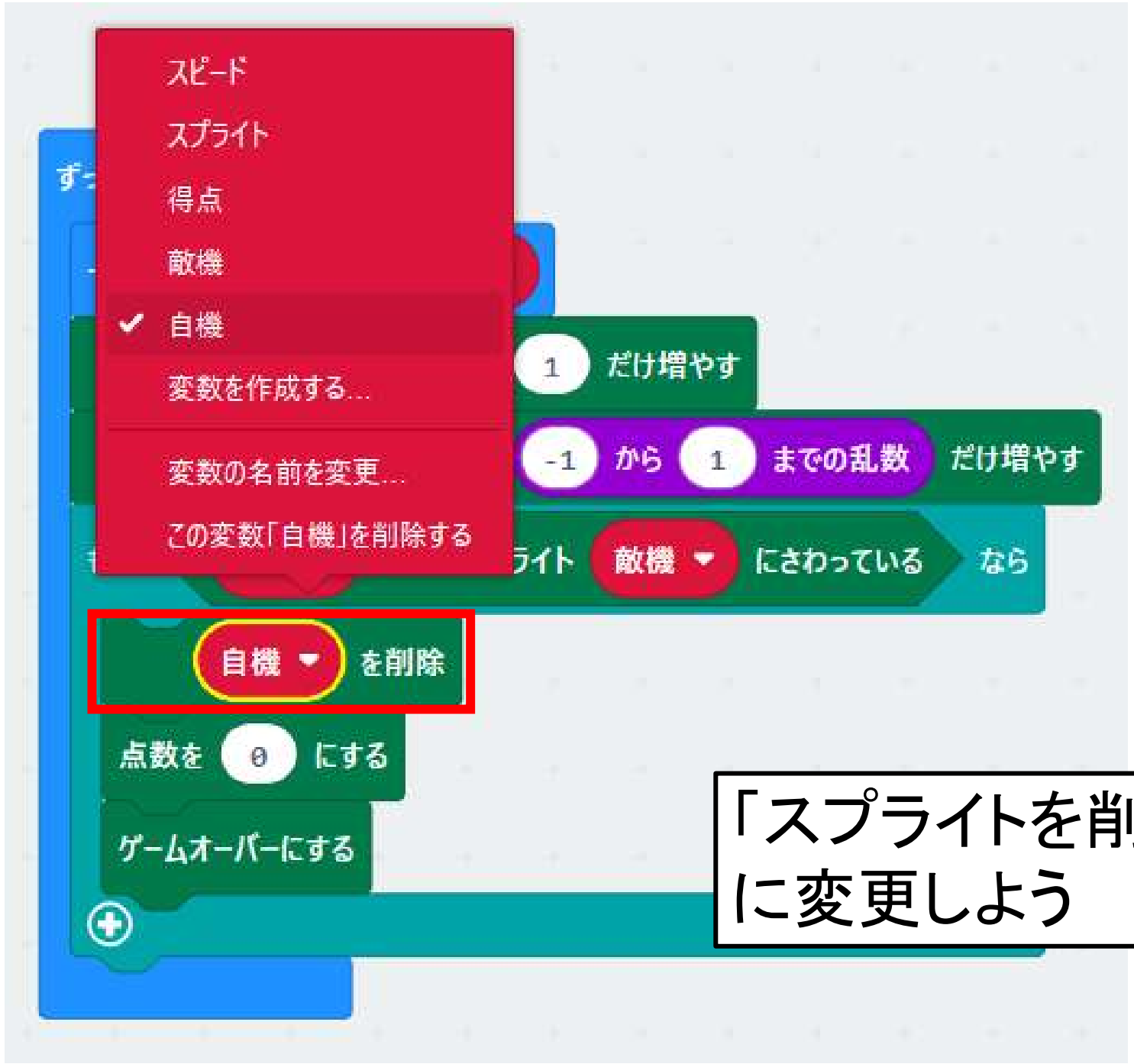
「変数」から「敵機」を持ってきて、
図のように組み合わせよう


無線
ループ
論理
変数
計算
高度なブロック
関数
配列
文字列
ゲーム
その他
画像
入出力端子
シリアル通信
ゲーム
スプライトを作成 X: 2 Y: 2
スプライト を削除
スプライト が削除済み
ゲームエンジンからスプライトを削除します。スプライトは画面に表示されたり、他のスプライトと連動したりすることはなくなります。
スプライト を
スプライト 方向転換 右 に 45 °
スプライト の X を 1 だけ増やす
スプライト の X に 0
スプライト の X
ずっと
一時停止（ミリ秒） スピード
敵機 の Y を 1 だけ増やす
敵機 の X を -1 から 1 までの乱数 だけ増やす
もし 自機 が他のスプライト 敵機 にさわっている なら
スプライト を削除
「ゲーム」から「スプライトを削除」を
持ってきて、図のように組み合わせよう

「ゲーム」から「点数を0にする」を
持ってきて、図のように組み合わせよう

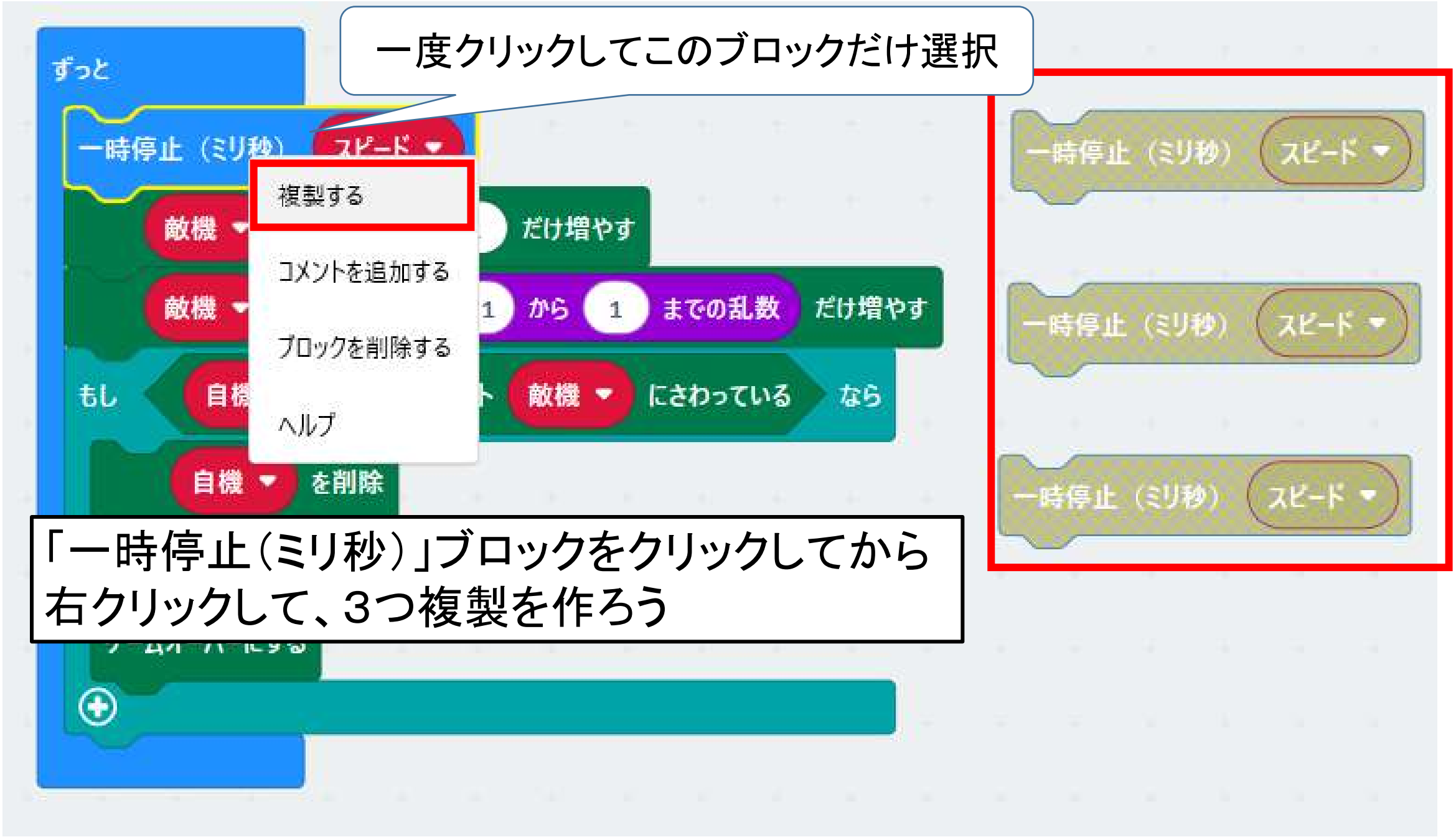


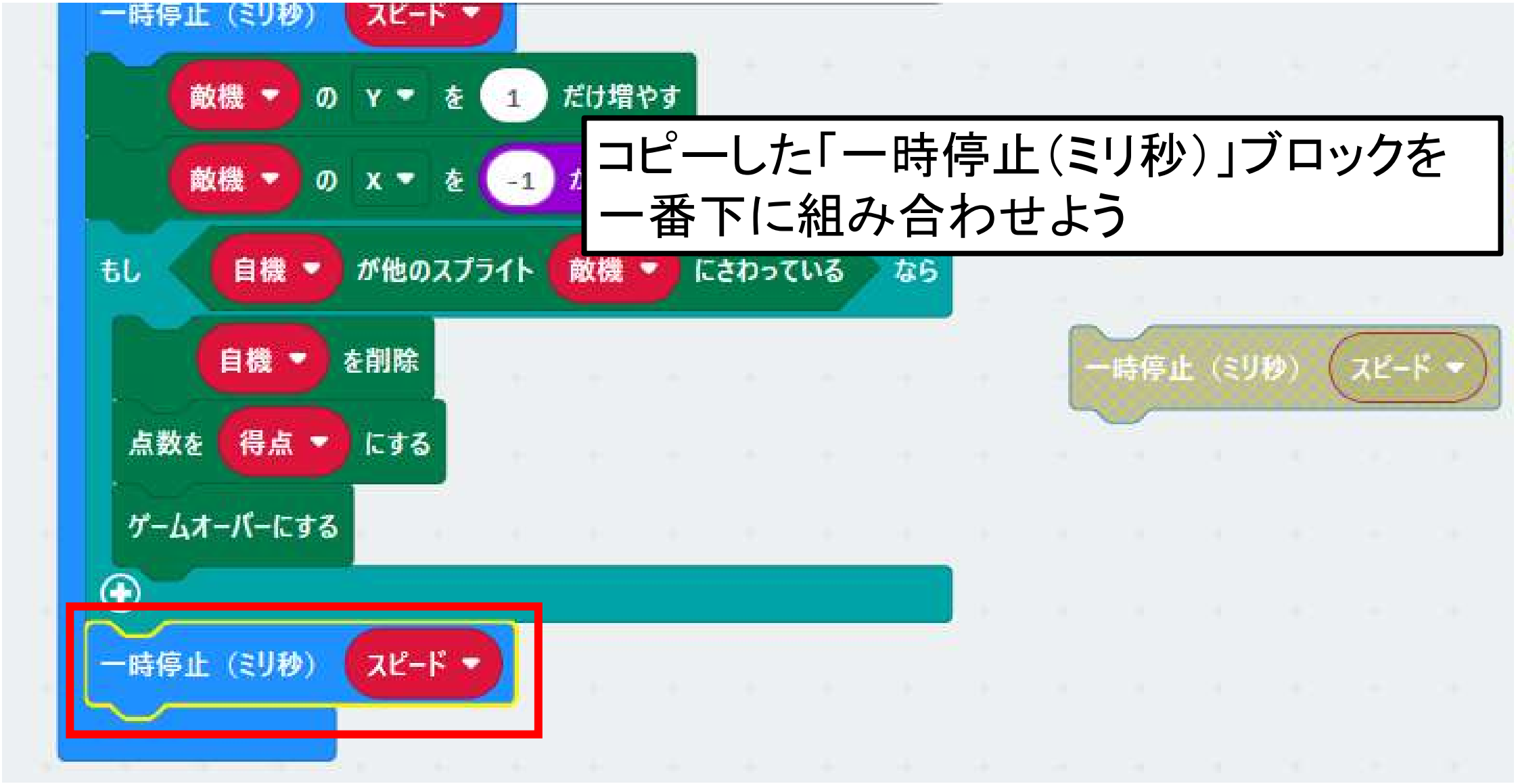
「ゲーム」から「ゲームオーバーにする」を
持ってきて、図のように組み合わせよう



「スプライトを削除」を「自機を削除」に変更しよう

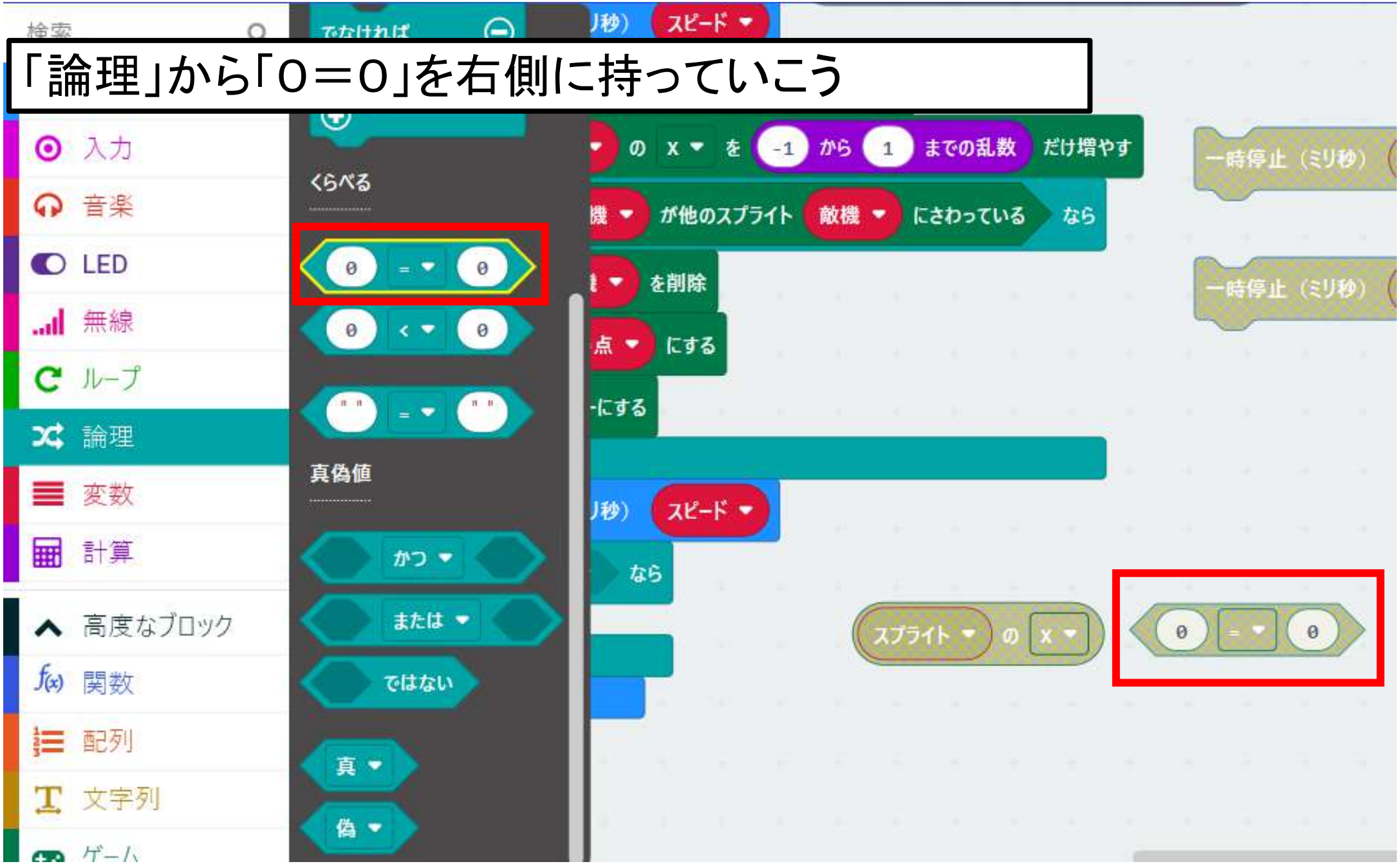

検索...
基本
入力
音楽
LED
無線
ループ
論理
変数
計算
高度なブロック
関数
配列
文字列
変数
変数を追加する...
スピード
スプライト
得点
敵機
自機
この変数の値を返します。
変数 スプライト を 0 にする
変数 スプライト を 1 だけ増やす
「変数」から「得点」を持ってきて、
「点数を0にする」に組み合わせよう
ずっと
一時停止（ミリ秒） スピード
敵機 の Y を 1 だけ増やす
敵機 の X を -1 から 1 までの乱数 だけ増やす
もし 自機 が他のスプライト 敵機 にさわっている なら
自機 を削除
点数を 得点 にする
ゲームオーバーにする


一度クリックしてこのブロックだけ選択
ずっと
一時停止（ミリ秒）
スピード
複製する
コメントを追加する
ブロックを削除する
ヘルプ
敵機
だけ増やす
敵機
1
から
1
までの乱数
だけ増やす
もし
敵機
にさわっている
なら
自機
を削除
一時停止（ミリ秒）
スピード
一時停止（ミリ秒）
スピード
一時停止（ミリ秒）
スピード
「一時停止（ミリ秒）」ブロックをクリックしてから
右クリックして、3つ複製を作ろう

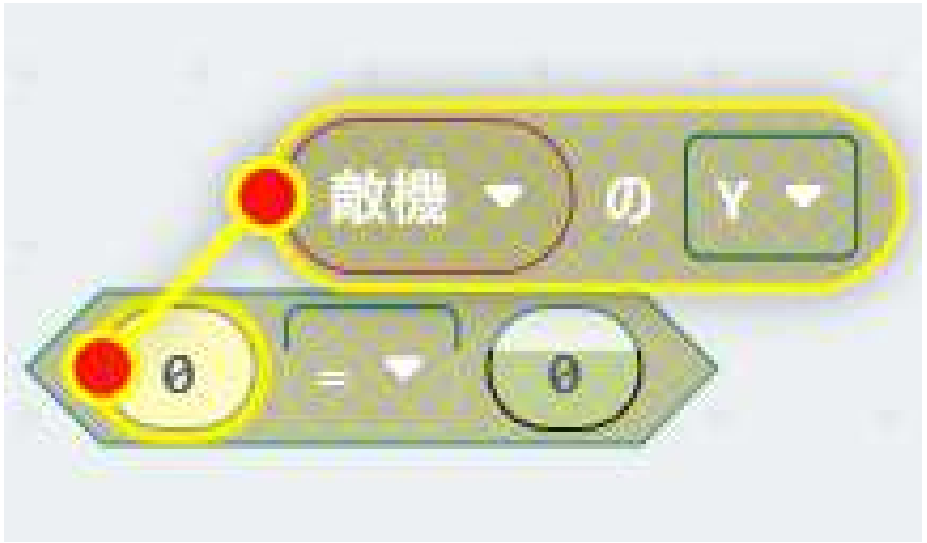

一時停止（ミリ秒） スピード
敵機 の Y を 1 だけ増やす
敵機 の X を -1
もし 自機 が他のスプライト 敵機 にさわっている なら
自機 を削除
点数を 得点 にする
ゲームオーバーにする
一時停止（ミリ秒） スピード
一時停止（ミリ秒） スピード
コピーした「一時停止（ミリ秒）」ブロックを
一番下に組み合わせよう

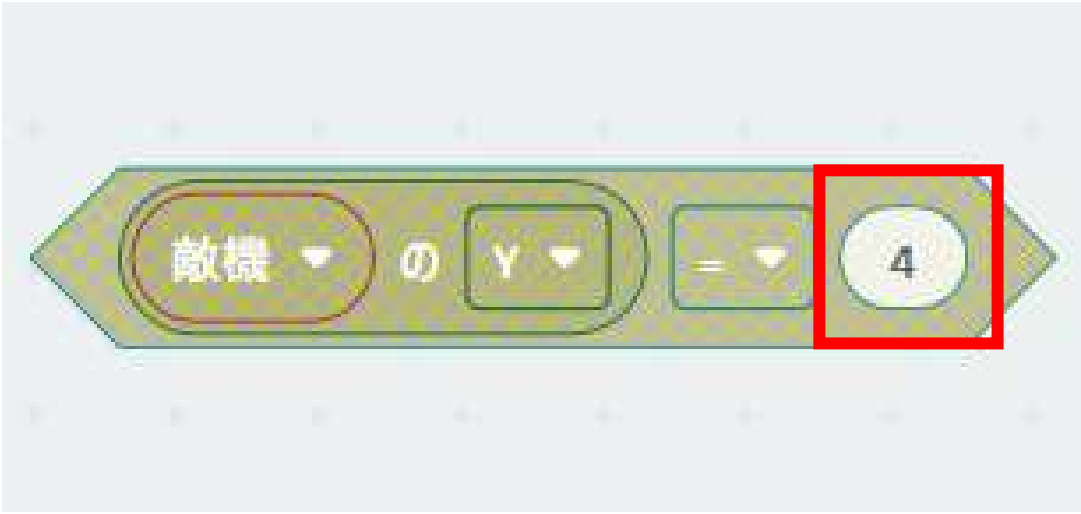
「論理」から「もし真なら」を
持ってきて、図のように組み合わせよう


「ゲーム」から「スプライトのX」を右側に持っていこう
無線
論理
変数
計算
高度なブロック
関数
配列
文字列
ゲーム
その他
画像
入出力端子
シリアル通信
制御
スプライトを作成 X: 2 Y: 2
スプライト を削除
スプライト が削除済み
スプライト を 1 ドット進める
スプライト 方向転換 右 に 45 °
スプライト の X を 1 だけ増やす
スプライト の X に 0 を設定する
スプライト の X
スプライトのプロパティを取得します。
スプライト
一時停止（ミリ秒） スピード
を 1 だけ増やす
敵機 の X を -1 から 1 までの乱数 だけ増やす
もし 自機 が他のスプライト 敵機 にさわっている なら
自機 を削除
点数を 得点 にする
ゲームオーバーにする
一時停止（ミリ秒） スピード
もし 真 なら
スプライト の X


「論理」から「0＝0」を右側に持っていこう
入力
音楽
LED
無線
ループ
論理
変数
計算
高度なブロック
関数
配列
文字列
くらべる
0 = 0
0 < 0
" " = " "
真偽値
かつ
または
ではない
真
偽
スピード
の x を -1 から 1 までの乱数 だけ増やす
が他のスプライト 敵機 にさわっている なら
を削除
にする
スピード
なら
スプライト の x
0 = 0
一時停止（ミリ秒）
一時停止（ミリ秒）

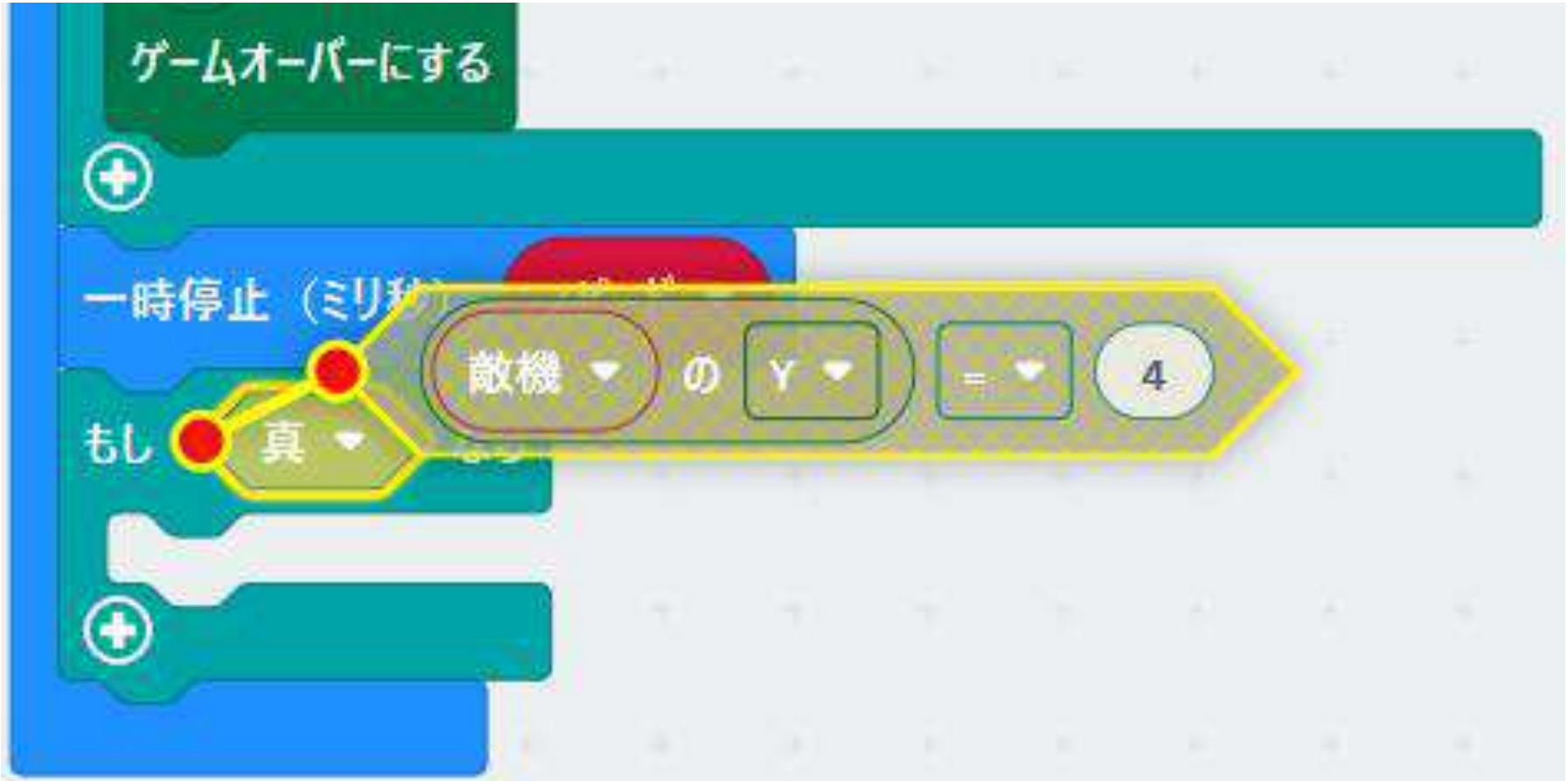
「スプライトのX」ブロックを「敵機のY」に変更しよう





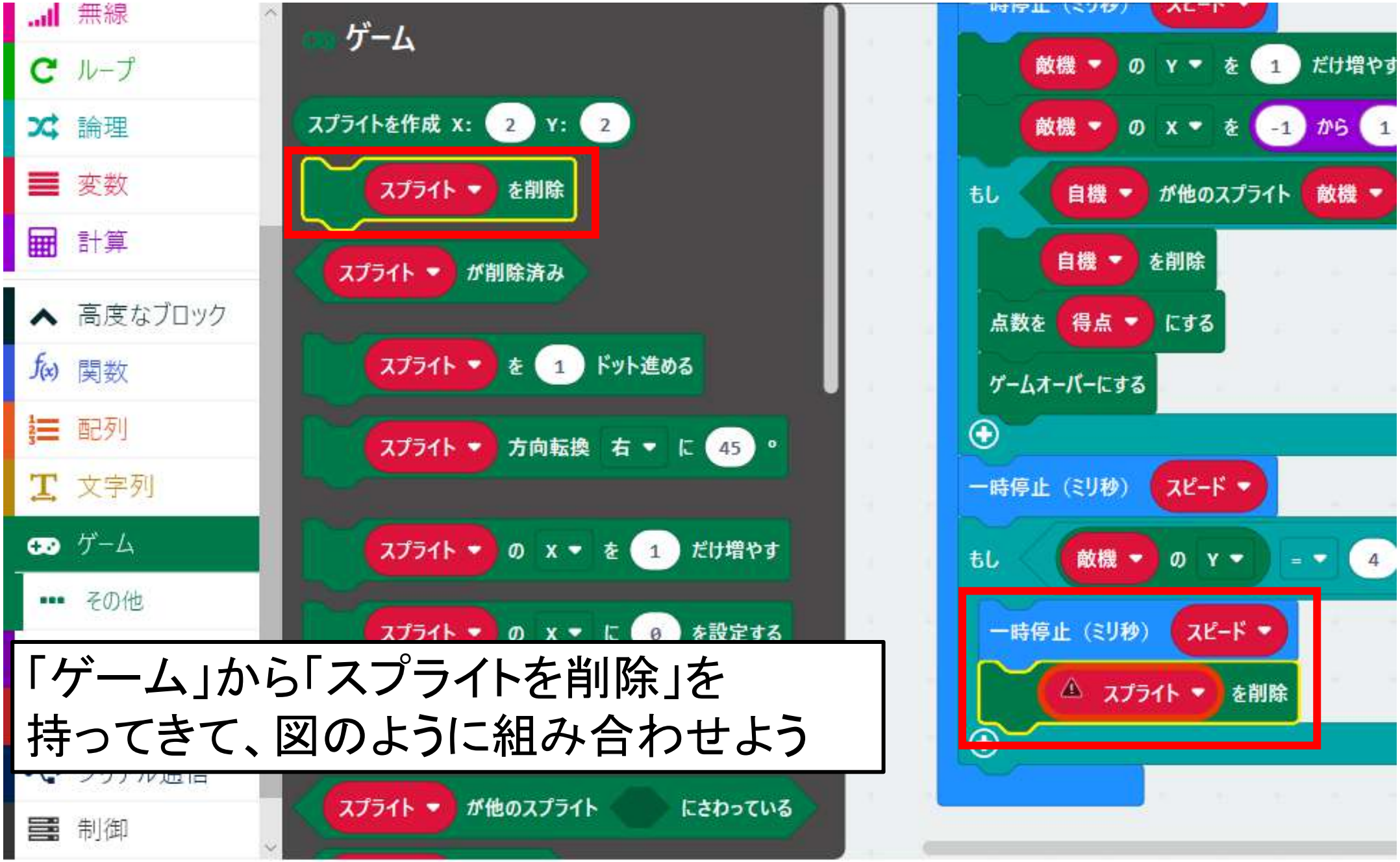
「敵機のY」と「０＝０」ブロックを合体させよう

右の数字を4に変更して、「敵機のY＝4」にしよう

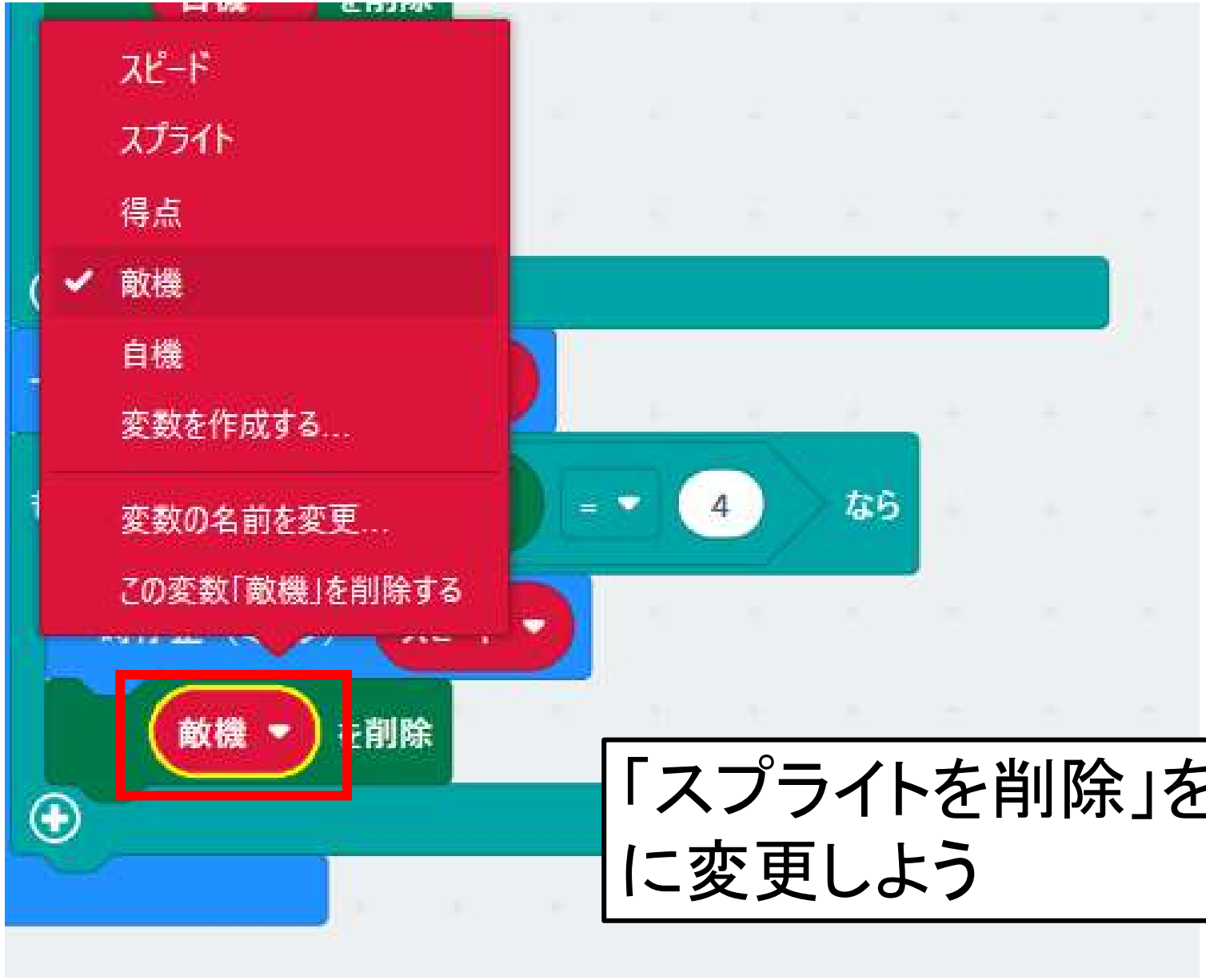


合体させたブロックを「もし真なら」ブロックに
組み合わせよう

コピーしておいた「一時停止（ミリ秒）」ブロックを
図のように組み合わせよう

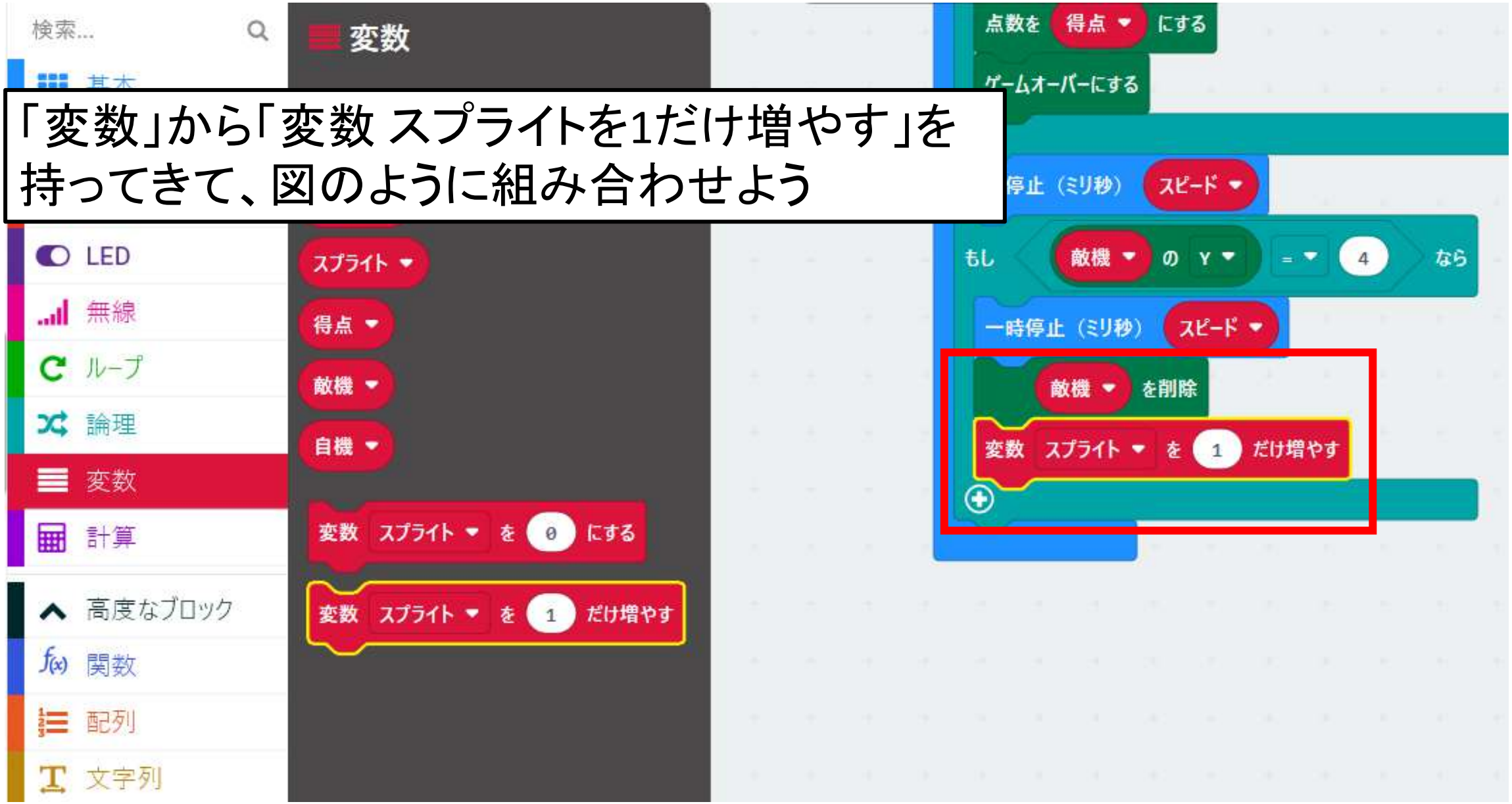


「ゲーム」から「スプライトを削除」を
持ってきて、図のように組み合わせよう



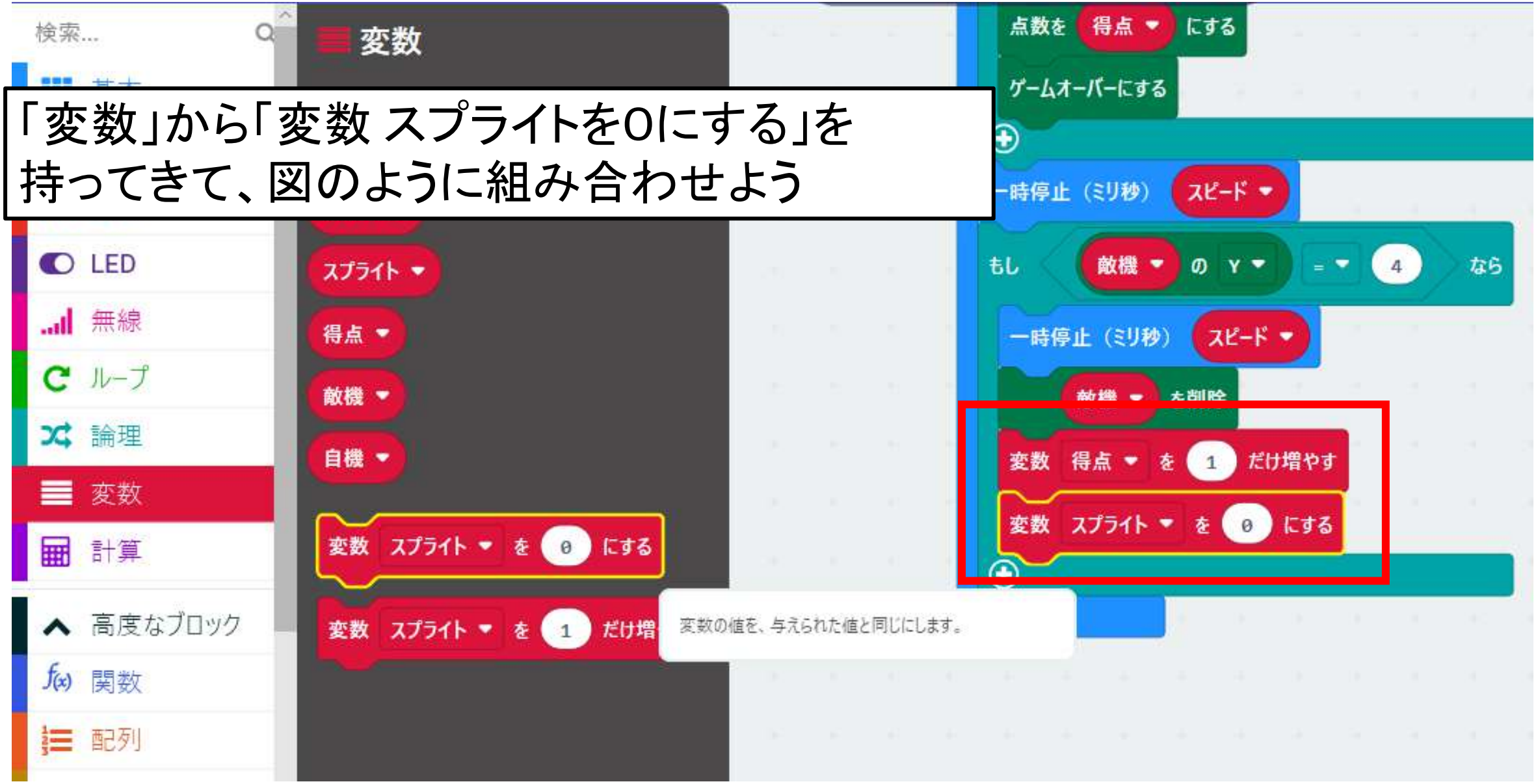
「スプライトを削除」を「敵機を削除」に変更しよう

「変数」から「変数 スプライトを1だけ増やす」を持ってきて、図のように組み合わせよう

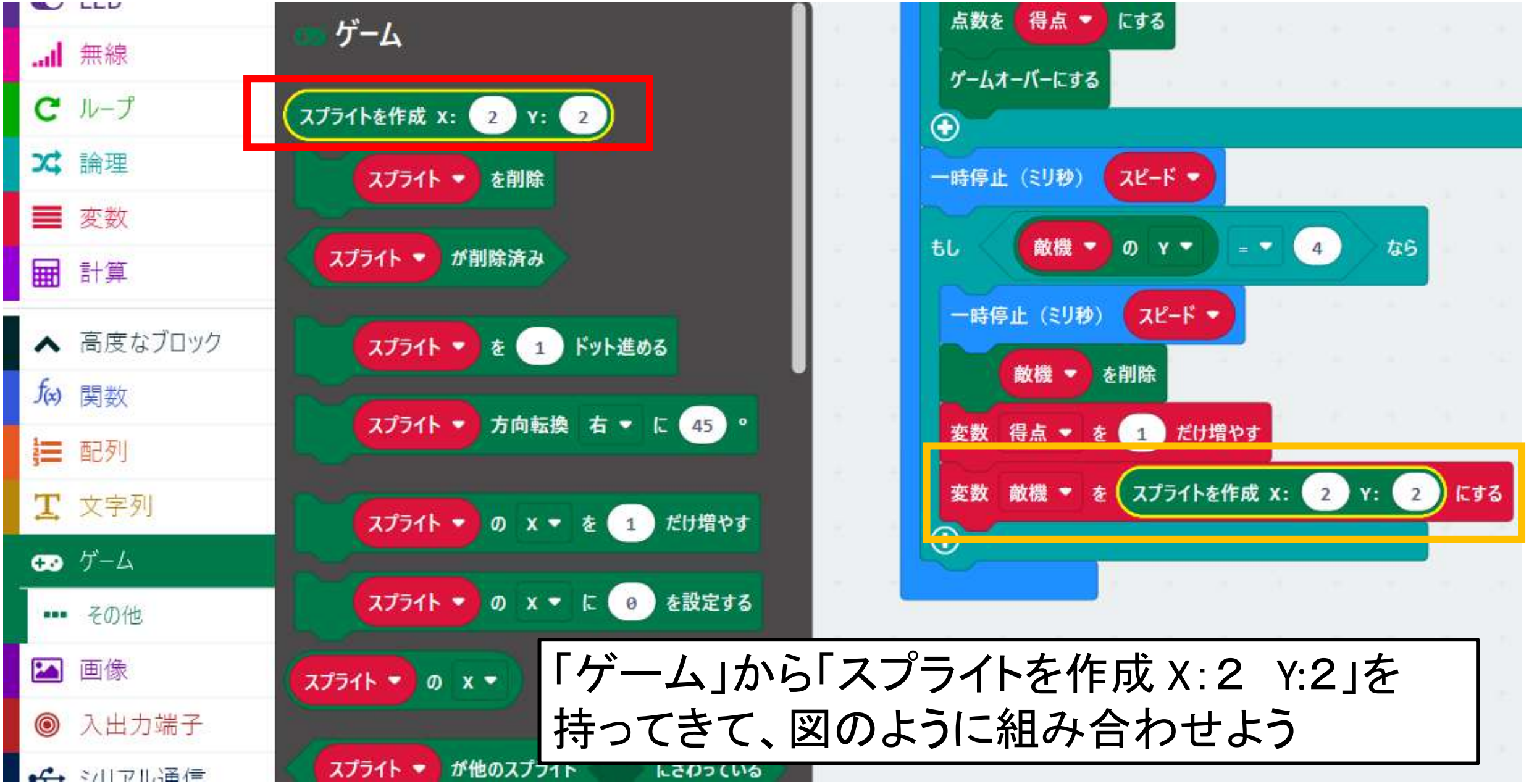


「スプライト」を「得点を1だけ増やす」に変更しよう

「変数」から「変数 スプライトを0にする」を
持ってきて、図のように組み合わせよう

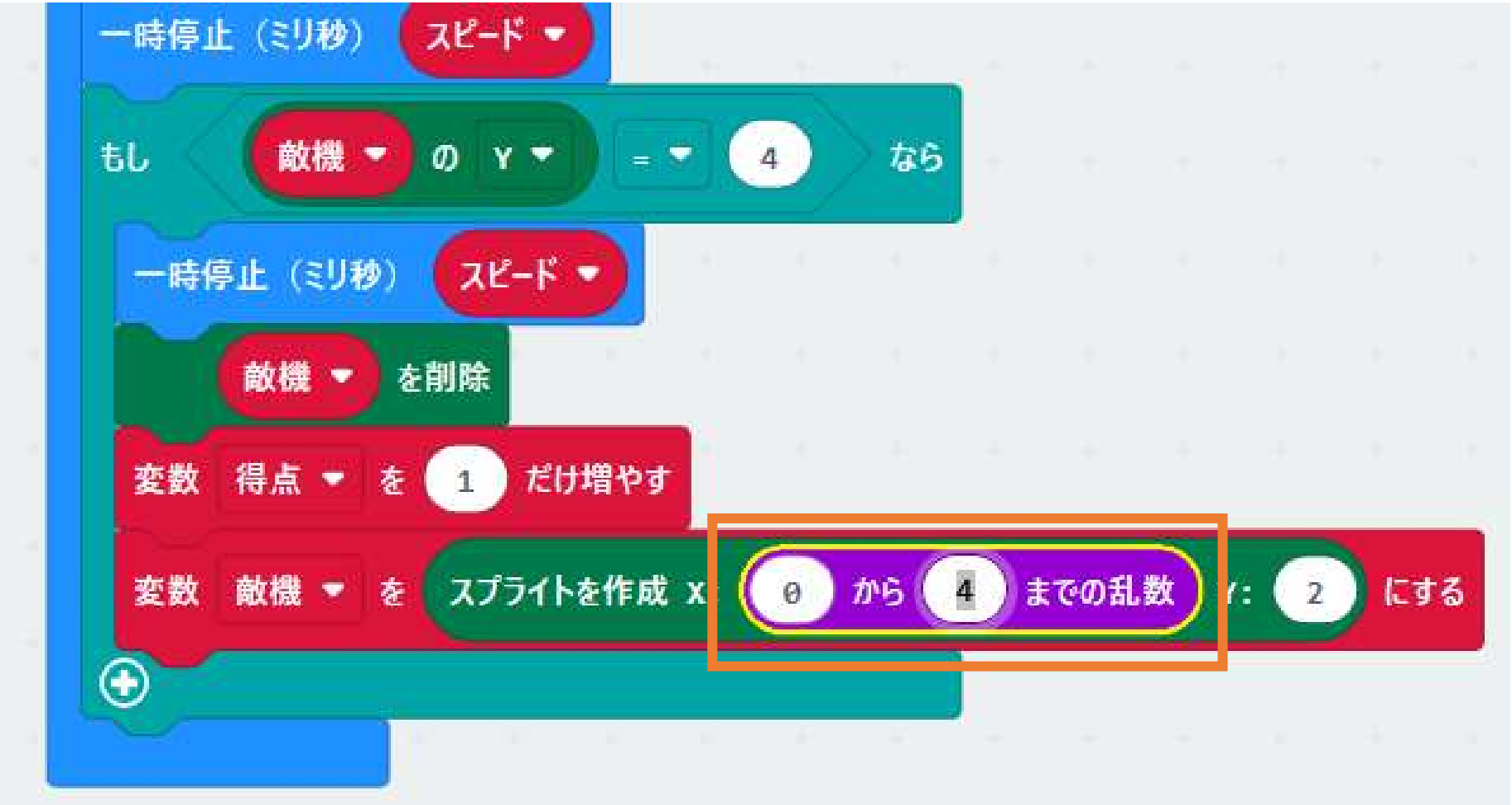


「スプライト」を「敵機をOにする」に変更しよう



「ゲーム」から「スプライトを作成 X:2　Y:2」を
持ってきて、図のように組み合わせよう

「計算」から「0から10までの乱数」を
持ってきて、図のように組み合わせよう



「0から10までの乱数」を「0から4までの乱数」に変更しよう

コピーしておいた最後の「一時停止（ミリ秒）」を
図のように組み合わせて完成！

実際に動かしてみよう！