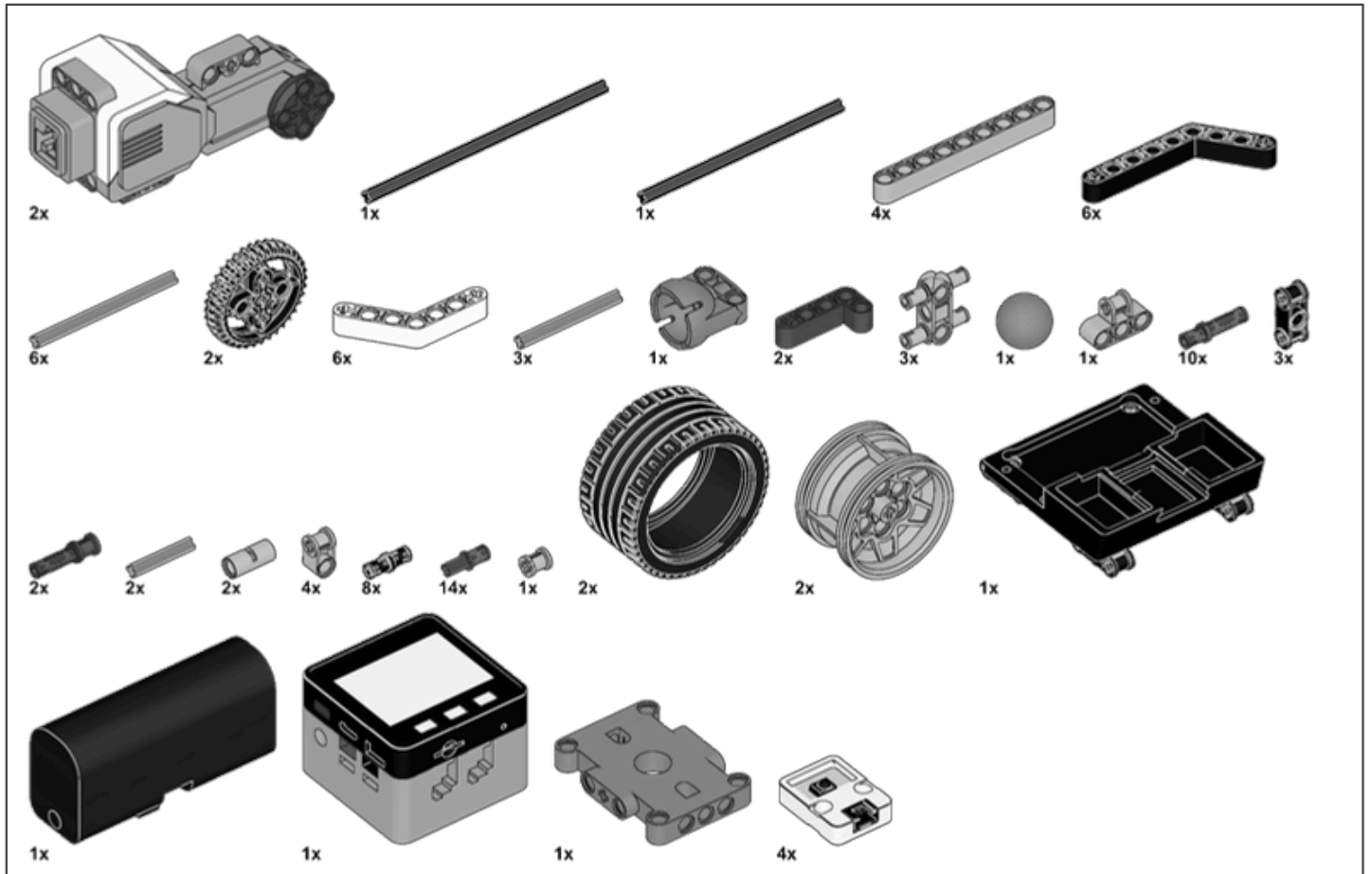


サッカーロボットをつくろう

Create Soccer Robot



for e-ピカ・かがわ

Version.202510-018

with **hrepcode**

kusuwata

もくじ

P

プログラミングをはじめよう

M

モータをまわしてみよう

B

ボールセンサを使ってみよう

V

変数を使ってみよう

C

コンパスセンサを使ってみよう

S

サッカープログラムを完成させよう

約束を守ってね

- 先生の言うことを **きこう**
- 名札を **必ず**つけよう
- 館内では **走らない**ようにしよう
- 教室では **食べたり飲んだりしない**ようにしよう
- 自分のロボットとタブレットを **大事に扱おう**
- **整理整頓**をがんばろう
- **時間を守ろう**
- 人のパソコンには **いたずらしない**ようにしよう
- 得意な子は他の子に **教えてあげよう**
- **同じ教室の仲間を大切にしよう**

分からないことがあったら、いつでも先生に聞いてね！



プログラミングを はじめよう

- .01 QRからプログラミングをはじめよう
- .02 ブロックを追加してみよう
- .03 数値や文字を入力しよう
- .04 画面に文字を表示してみよう
- .05 プログラムをロボットに書き込もう

QRからプログラミングをはじめよう

P プログラミングをはじめよう




hrepcode (eとぴあ版)

<https://etp.kusuwata.com>



ロボットを「ブロック」で簡単に動かせるように作られたアプリです。

画面右下の  を押すと、QRコードスキャナが起動します。ここからロボットやタブレットと接続したり、このテキストからプログラムを開いたりすることができます。



設定に使うQRコード

ユーザー：
default_robot



iPadを誰が使うのかを設定します。

APIKEY：
F987ECC1



iPadとロボットを接続します。

プログラムをひらくQRコード

QRを読み込んで
プログラミングを開始



自由に編集することができるプログラムが開きます。
もう一度同じQRコードを読めば
あなたが作ったプログラムの最新版が開きます。

QRを読み込んで
動きをみてみよう



完成したプログラムが開きます。
数値や時間は変更できますが、あたらしいブロック
を追加したり消したりはできません。
もう一度同じQRコードを読んでも
変更がリセットされたプログラムがひらきます。



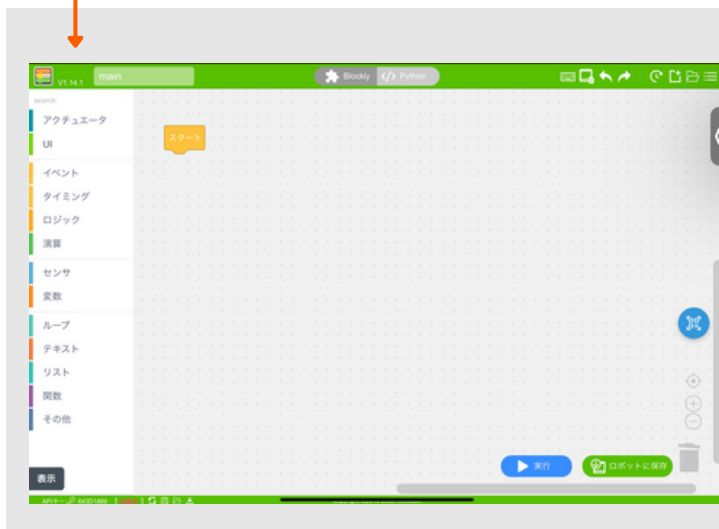
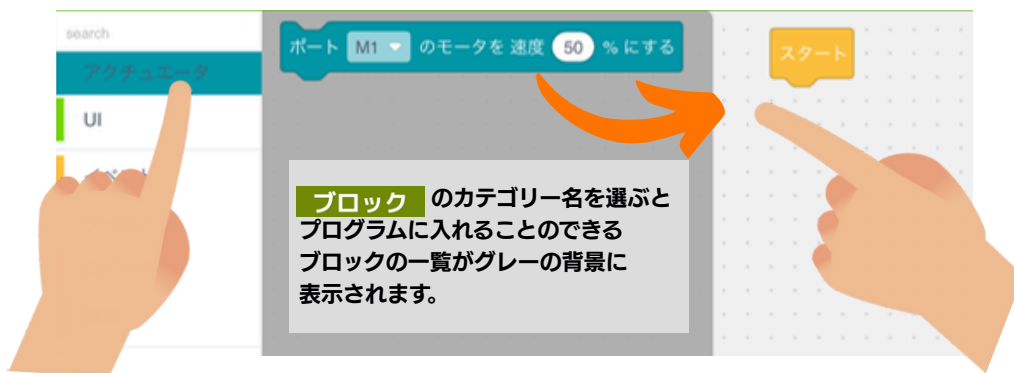
ブロックを追加してみよう

P

プログラミングを
はじめよう

画面は大きく3つのエリアに分かれています。

プログラムを作るときには **ブロック** エリアから使いたいブロックを選び
プログラミングエリア にならべてプログラミングします。



画面を広く使って
プログラムを書きたいときには

画面設定エリア 「隠す」ボタンで
を隠すことができます。

再表示するには「表示」を
押してください。



数値や文字を入力しよう

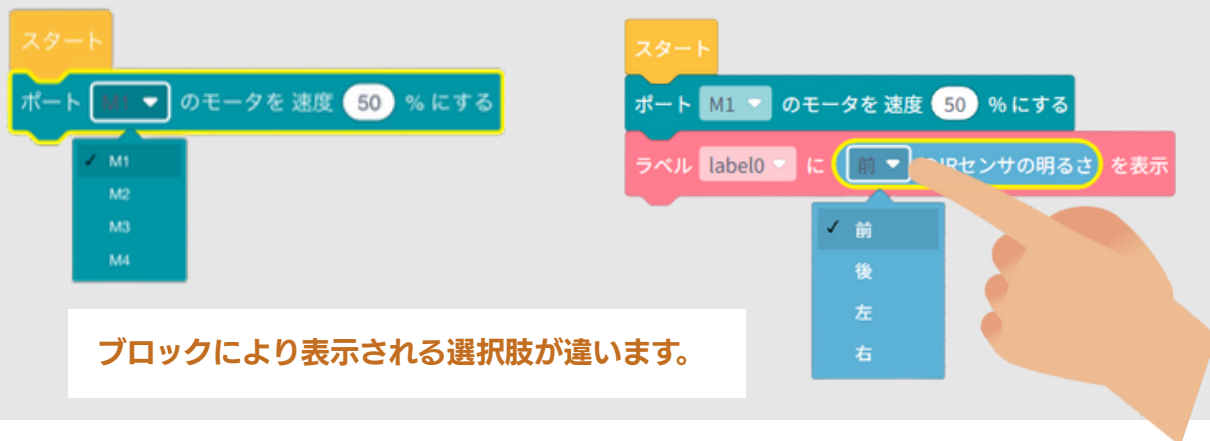
P

プログラミングを
はじめよう

数値やポートを変更しよう。

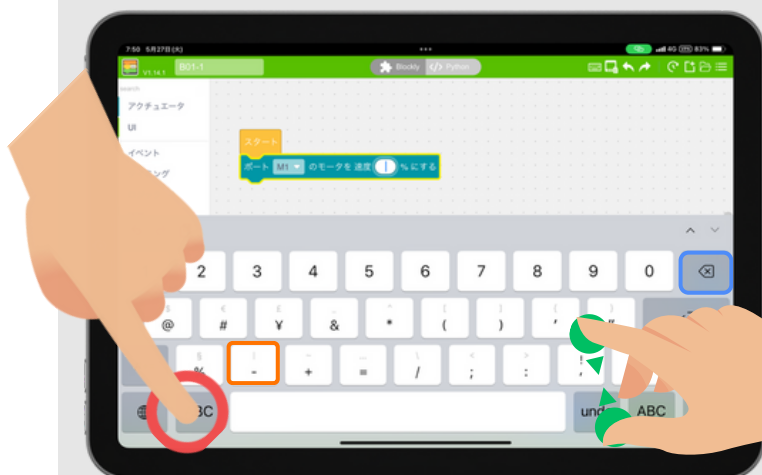
プログラムで使うブロックには自分で変えられる部分があります。
例えば 速度、パワー、角度、明るさ、ポート・・・など。
これらを変更するにはいくつかの方法があるので紹介します。

ブロック上に表示される候補をタップして選ぶ



キーボードでの入力

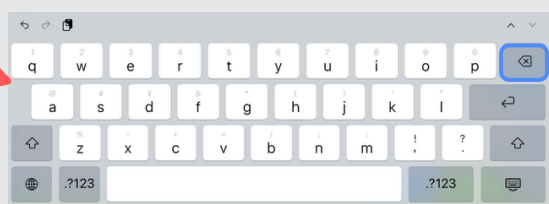
設定されている値が「数字」や「文字」のときにはキーボードが表示されます。
数字の入力には小数点を使うことも可能です。



数値を入力するとき、キーボード表示上で
2本の指を広げる・せばめる 操作をすれば
キーボードの大きさを切り替えられます。



ABC または .?123 キーを押すと、
数字の入力と文字の入力を切り替えられます。



1 文字削除
文字を消すときは
このキーを使うよ。

マイナス記号は
大きいキーボードの
ときだけ入力できるよ。



画面に文字を表示しよう

P

プログラミングを
はじめよう

ロボットの中にある画面の、どこに・どんな文字や図形を表示するのかを画面設定エリア上で設定することができます。

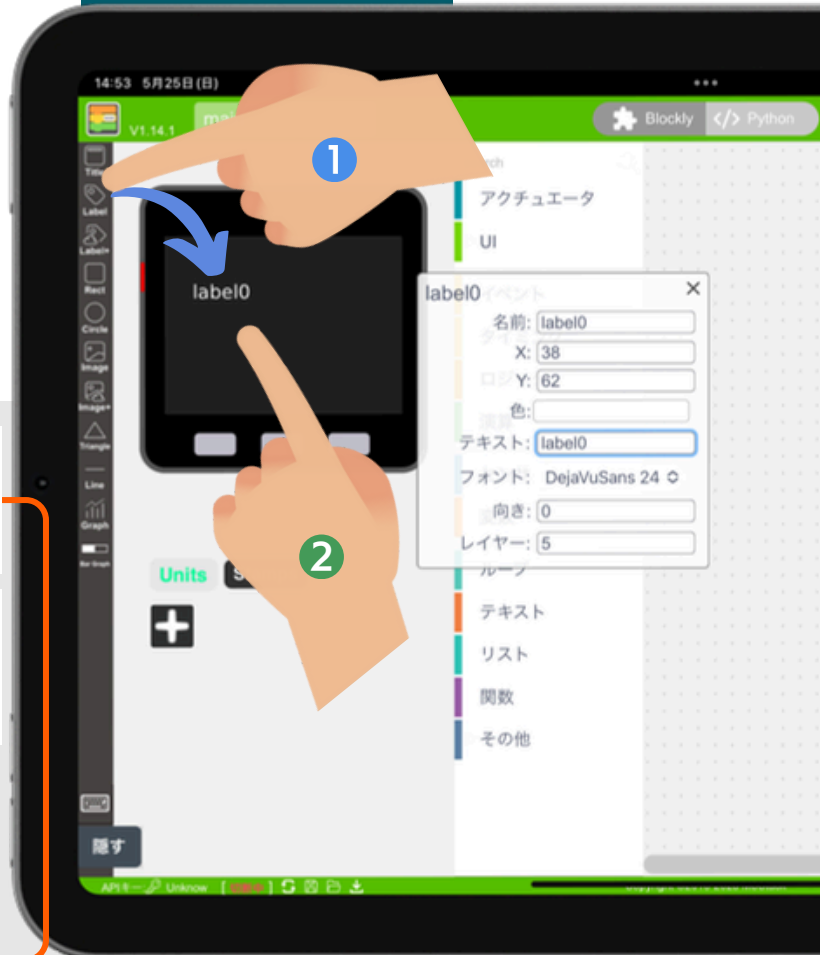
①ラベルを追加する

画面に文字を表示したいときは、で囲んだ「Label」を指で押して、画面の上までスライドしてください。

②ラベルの設定を変える

文字の大きさや色・向きなどはさきほど配置した「Label0」をタップすれば設定できます。

画面設定エリア



ラベルの**名前**を変えられます。
後で分かりやすい設定にしましょう。

文字の色を変えることができます。
色を選んだら「OK」
を押してください。



label0

名前: label0

X: 62

Y: 49

色:

テキスト: label0

文字/サイズ: Default

向き: Default

レイヤー: Default Small

Ubuntu-C

Comic

DejaVuSans 18

DejaVuSans 24

DejaVuSans 40

DejaVuSans 56

DejaVuSans 72

Unicode 24

表示する**文字**を入力します。
Unicode24以外は
英数字・記号しか入力できません。

表示する**文字**の**フォント**や**大きさ**を
設定します。

DejaVuSans 18
DejaVuSans 24
DejaVuSans 40
DejaVuSans 56
DejaVuSans 72



プログラムをロボットに書き込もう

P プログラミングを
はじめよう

作ったプログラムをすぐにロボットで動かすことも、ロボット内部に保存しておくこともできます。

プログラムを実行しよう

▶ 実行 ボタン

今画面に表示されているプログラムをすぐにロボット上ではしらせるよ。プログラムを変更して実行すると、すぐ最新のプログラムに書き換わります。

電源が落ちたら
プログラムは削除されます。

すぐ動き出すから、ロボットを机から落としたりしないように気をつけて！



不安なときは
立てて置くと
安心だよ。

📁 ロボットに保存 ボタン

今画面に表示されているプログラムをロボットの中に保存します。


App ボタンを押すと、
AppList がひらきます。

ここでプログラムを選択して開くことが可能だよ。



この方法でロボットに保存したプログラムは次の授業でも同じように開くことができます。

ロボットを停止する方法

ロボット右側のボタン  をおしてプログラムの実行を止めよう。



電源が切れなくなったら

DC Port-B
を抜いてください



ロボットを充電する方法

USBケーブルをさせばロボットの電源が自動でOFFになり、充電が開始されるよ。

USBタイプCケーブル
で充電が可能です。



電池残量

4つの光で表示。
充電中はLEDの1つが点滅します。

電源ON

または電池残量の確認ができます。



モータを まわしてみよう

- .01 速度と向きを変えてみよう
- .02 いろんな動きを試してみよう
- .03 時間を指定してコースをたどらせよう

速度と向きを変えてみよう

M

モータを
まわしてみよう

モータ1がフルパワーでまわります。

MO1-1



QRを読み込んで
動きをみてください



説明

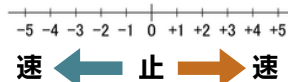
どのモータをまわす？「ポート」

どのモータを回転させるのかを決めます
ロボットの
左タイヤがM1
右タイヤがM2



モータをまわす「向きと速さ」

-100 ~ 100で
回転速度を指定します。

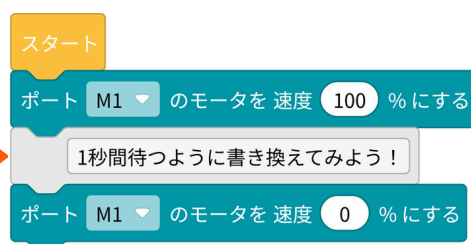


一の値が大きくなるほど 時計まわり に速く回る
+の値が大きくなるほど 反時計まわり に速く回る

モータを1秒間・時計回りにまわそう

ある動きをつづける「時間」を指定するには「待つ」ブロックをつかいます。
QRから読みこまれたプログラムを書き換えて、
モータが1秒間回転して停止するようにしてください。

MO1-2



QRを読み込んで
プログラミングを開始



説明



このブロックを使うことによって、
決めた時間待ってから、下のブロックを実行します。
待つ時間は小数を使って1秒より細かく調整できます。

算数

小学5年 【速さ・時間・距離】の単元で習う内容だよ。
次のページでいろんな動きを試しながら、その関係を見ていこう！

いろんな動きをためしてみよう

M

モータを
まわしてみよう

モータを自分が決めた時間まわすことができました。
これを左右2つのモータで実行すればロボットはどう動くでしょうか。

実際にロボットを動かしながら、**パワーと動きの関係**を考えてみよう。

MO2-1

スタート

動きを設定しよう

ポート M1 のモータを速度 0 %にする

ポート **ここにブロックを追加しよう** %にする

5 秒待つ

5秒間たったら停止

ポート M1 のモータを速度 0 %にする

ポート M2 のモータを速度 0 %にする

QRを読み込んで
プログラミングを開始



モータのブロックは
「アクチュエータ」のなかにあるよ。

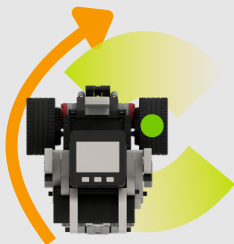


直進



まっすぐ前に進む

回転



右タイヤを軸に回転

カーブ



ロボットの中心を軸に回転



ゆるやかに右カーブ



時間を指定してコースをたどらせよう

M

モータを
まわしてみよう

次は直角のコースをたどってみましょう。

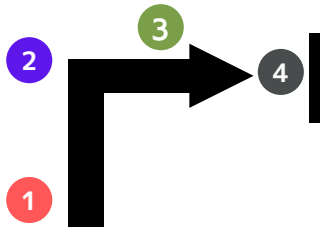
決められたコースをたどるために動きを4つに分けて書いてみよう

- どのような「動き」を
- どのような「順番」で

組み合わせるとコースをたどることができるでしょうか。
動きを四角の中に入れて書き込みながら考えてみよう。

右の枠から各番号にあう
動きを選んで
書き込んでみよう。

回転 / 前進 / 止まる / 前進



1	2	3	4
---	---	---	---

どのような速度と時間を入れたら、上の **4つの動き** をできるのか
値を決めてロボットを走らせよう！

MO2-2

スタート

ポート M1 のモータを速度 % にする

ポート M2 のモータを速度 % にする

秒待つ

ポート M1 のモータを速度 % にする

ポート M2 のモータを速度 % にする

秒待つ

ポート M1 のモータを速度 % にする

ポート M2 のモータを速度 % にする

秒待つ

ポート M1 のモータを速度 % にする

ポート M2 のモータを速度 % にする

QRを読み込んで
プログラミングを開始



M1	M2	秒
M1	M2	秒
M1	M2	秒
M1	M2	秒



B

Ball

ボールセンサを 使ってみよう

- .01 ボールセンサの値を見てみよう
- .02 ボールセンサの値で文字色を変えよう
- .03 ボールセンサの値で動きを変えよう

ボールセンサの値を見てみよう

B ボールセンサを使ってみよう

ボールセンサは、目に見えない「赤外線（せきがいせん）」を使って、ボールの場所を見つけることができます。

競技用のボールは赤外線を出しています。
センサはその赤外線の強さを感じとり「どれくらい近くにボールがあるか」が分かります。



赤外線の強さを画面に表示してみよう

B01-1



QRを読み込んで動きをみてみよう



プログラムを動かすと、前のボールセンサが感じた赤外線の強さが表示されます。
センサの数字が大きいほど、赤外線が強くボールが近くにあることを意味します。

下の図にボールをおいて、センサで読んだ数字をメモしてみよう。
いろいろな位置で試して、どんなふうになるか見てみよう！

画面に表示された数値をみてね

次ページで新たに登場したブロックの解説をするよ！



ボールセンサの値で文字色を変えよう

B

ボールセンサを
使ってみよう

ラベル

文字や数字をロボット上の画面に表示するよ



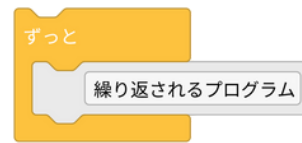
どのラベルに文字を表示するか変更できる

表示できるのはどんな文字？

- ・英語の文章
- ・数字
- ・% &! ? () [] { } + - * / などの基本の記号

無条件ループ

中にいれたプログラムをずっと繰り返すよ



B01-1では
なぜボールセンサの数値を確認するのに
「無条件ループ」を使っていたのかな？

ボールのあり・なしで文字の色を変えよう。

B02-1



QRを読み込んで
プログラミングを開始



みどりの部分は自分でつくろう

ロボットからみてボールの赤外線が
明るいときには 赤く
暗いときには 青く

表示されている文字の色が変わります。

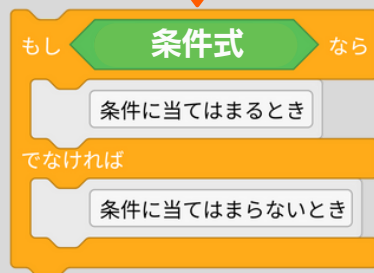
条件分岐とは

条件分岐とは、ある条件に

合てはまるとき (if)

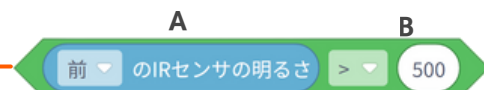
合てはまらないとき (else) に、

する動きを分けてプログラムするしくみです。



「不等式」とは

何と何をどう比べるのか、決めるためのもの。
多くの比べ方があるので、下に例を示します。



$A=B$	A と B が 等しい	とき
$A \neq B$	A と B が 等しくない	とき
$A < B$	A が B より 小さい	とき
$A \leq B$	A が B と 同じか小さい	とき
$A > B$	A が B より 大きい	とき
$A \geq B$	A が B と 同じか大きい	とき

数学

中学1年 【不等式】の単元で習う内容だよ。

2つの数字を比べたり、条件を満たす値を計算したりするよ。

ボールセンサの値で動きを変えよう

B

ボールセンサを
使ってみよう

ボールの **あり・なし** で 実際にロボットを動かそう。

条件分岐でロボットを動かそう。

文字の色が変わるプログラムに、動きを追加しよう。

「**ボールがある**とき」に**前進**

「**ボールがない**とき」に**回転**させよう。

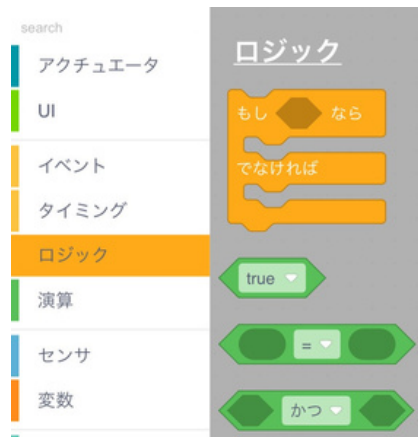
B03-1



QRを読み込んで
プログラミングを開始



条件分岐のブロックは
「ロジック」のなかにあるよ。



モータのブロックをすべて追加したら**速度も変更**しよう。





変数を使ってみよう

- .01 変数を使って数をカウントしよう
- .02 変数でボールの左右の位置を表そう
- .03 変数でモータの速度を計算しよう

変数を使って数をカウントしよう

V

変数を使ってみよう

変数を使って、ボタンが押されたときに表示する数字を変えるよ。

変数ってなに？

変数（へんすう）は、いろんな情報を入れておける「入れもの」のようなもの。
{ 数字 / 文字 / はい・いいえ } など、さまざまな情報を入れることができます。

今回は、**count** という変数に数字を入れて使っています。
まだ決まっていない数字でも、変数に名前をつけておけば、
あとから入れ変えたり、計算したり、よび出したりできます。



```
スタート
count を 0 にする
ずっと
もし ボタン A が クリックされた である なら
  count を 1 増やす
でなければ
もし ボタン B が クリックされた である なら
  count を 0 にする
でなければ
もし ボタン C が クリックされた である なら
  count を -1 増やす
でなければ
ラベル label0 に count を表示
```

QRを読み込んで動きを見てみよう



押されたボタンによって

それぞれ **count** に

増やす

入れ替え

減らす

処理をしているね。



count に設定した数字を呼び出して表示

VO1-1

このプログラムで使われている2つの変数設定ブロック

```
count を 0 にする
```

変数の中に入っている値を、設定した値に置き換えてしまうよ。

これまで数字を入れていた変数には数字を文字を入れていた変数には文字を

といったように、同じ種類の値同士で変更してね。

```
count を 1 増やす
```

変数の値を増やしたり減らしたりするにはこのブロックを使うよ。

例えば、
1増やしたいときは 1 を入れる
1減らしたいときは -1 を入れるだけ。

増やしたい数や減らしたい数をプラスやマイナスの数字で指定してね。

count を count + 1 にする と count を 1 増やす では同じ処理・同じ結果になるよ！



変数でボールの左右の位置を表そう

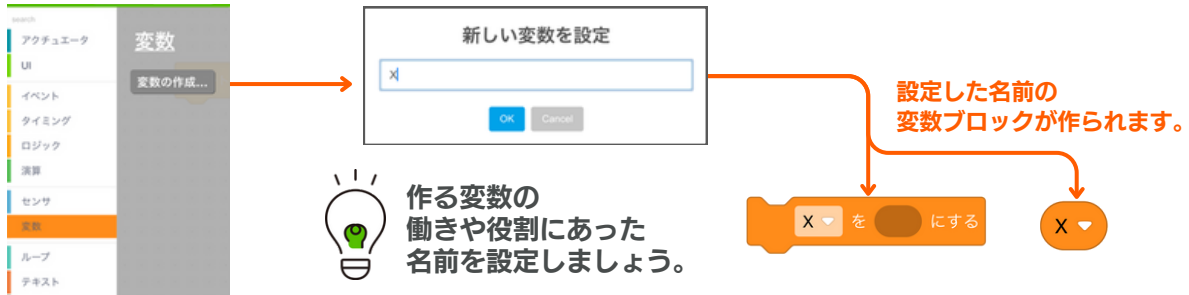
V

変数を使ってみよう

変数「X」を定義して、**ボールの左右の位置**をラベルに表示しよう。

変数のしくみや役割はすこし感じられたかな？
では実際に変数を使ってプログラムを書こう。

変数の作り方



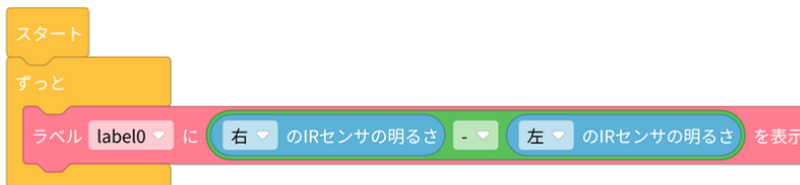
変数を使わずにつくられたプログラムが開くので一度実行してみよう。

次に、**変数「X」**を使った形に書き換えて、同じように動かか確認しましょう。

QRを読み込んでプログラミングを開始



V02-1



X の値とボールの位置の関係を確認しよう。

ロボットの左にいくほどXの値は…



くなる



ロボットの右にいくほどXの値は…



くなる

ボールが正面にくると…



ほぼ になる



変数でモータの速度を計算しよう

V

変数を使ってみよう

ボールの位置を示す変数「X」を使って、ボールの方を向こう。

変数Xを使って、ロボットがいつでもボールの方を向くようにするよ。

変数LとRは左右それぞれのモータの出力として使われているよ。

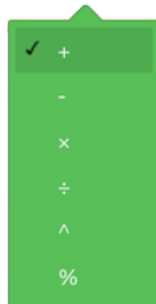
変数Lを参考にして
変数Rを追加して
パワーを計算させてみよう！

QRを読み込んで
プログラミングを開始



V03-1

```
スタート
ずっと
  X を 右 のIRセンサの明るさ - 左 のIRセンサの明るさ にする
  ラベル labelX に X を表示
  L を 0 + X にする
  ここで変数Rを計算しよう。
  ラベル labelL に L を表示
  ラベル labelR に R を表示
  ポート M1 のモータを速度 L %にする
  ポート M2 のモータを速度 R %にする
```



右モータのパワーを計算するには
どの計算をするといいかな？

ロボットはボールの方を向くようになったかな？

数値を調整して、ボールを「追いかける」ようにプログラムを変更しよう。

ロボットがボールを「追いかける」ためには、
「ボールの方を向く力」と、「前に進む力」を足せばいいよ。

理科

中学 【運動とエネルギー】の単元で習う内容だよ。
同じ方向の2つの力が合わさるとどうなるかなどを習うよ。



コンパスセンサを 使ってみよう

- .01 コンパスセンサの値を見よう
- .02 いつでも正面を向けよう
- .03 敵陣を向いてボールを追いかけてよう

コンパスセンサの値を見てみよう



コンパスセンサを
使ってみよう

コンパスセンサが示す「方角」を画面に表示しよう。

コンパスセンサの値を画面に表示させたよ。ロボットを回しながら値を確認しよう。

C01-1



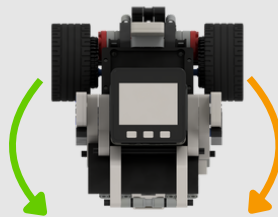
QRを読み込んで
動きをみてみよう



左回りに
回転させると…

値は

くなる



右回りに
回転させると…

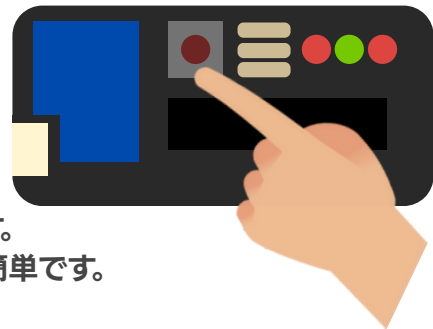
値は

くなる

正面の角度をリセットする

このロボットのコンパスセンサは簡単に「正面」の設定を変更することができます。

図に示したスイッチを押せば、値は0に変更されます。
この機能を活用すれば、敵陣の方角を指定するのも簡単です。



いつでも正面を向けよう



コンパスセンサを
使ってみよう

条件分岐とコンパスセンサの方位を使って 正面を向こう

「方位が -20度より 小さいとき」 → 「右回転」

「方位が 20度より 大きいとき」 → 「左回転」

どちらでもなければ「停止」するようにプログラムを書こう。

C02-1



QRを読み込んで
プログラミングを開始



復習のために、変数を使ったプログラムにしているよ。

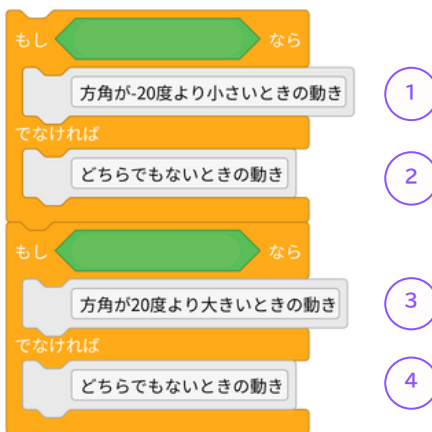
3行目で Z に 方位 (-180~180) の値を代入しているね。

- ① 「方位が -20度 より小さいとき」
- ② 「方位が 20度 より大きいとき」

①②この2つの状態をみきわめるには
どう条件式を設定したらいい？

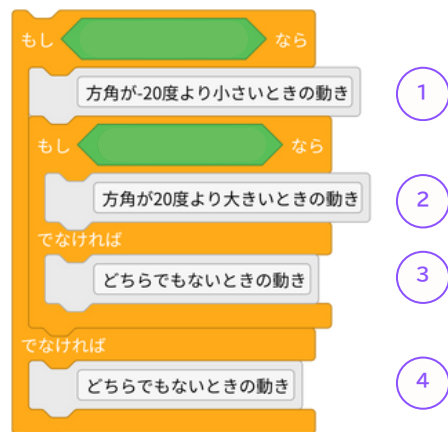
2つの条件分岐を組み合わせるときのよくある間違いプログラムを2つ作成しました。
もしこんなプログラムにしたらどんな問題が発生するかな？ 考えて書いてみよう。

例A



どんな問題が起こる？ _____

例B



どんな問題が起こる？ _____



敵陣を向いてボールを追いかけてよう



コンパスセンサを
使ってみよう

コンパスセンサとボールセンサのプログラムを組み合わせよう

QRコードからは「変数を使ってみよう」
最終回で作成したのと同じ
ボールを追いかけるプログラムが開きます。

QRを読み込んで
プログラミングを開始



「方角が-90度より小さければ」 → 「右回転」
「方角が 90度より大きければ」 → 「左回転」

するプログラムをプログラム内に書き加えて組み合わせてみよう。

Soccer

```
スタート
ずっと
  Z を 方角 (-180-180) にする
  ラベル labelZ に Z を表示
  もし Z < なら
    ポート M1 のモータを速度 50 %にする
    ポート M2 のモータを速度 -50 %にする
  でなければ
    もし Z > なら
      ポート M1 のモータを速度 -50 %にする
      ポート M2 のモータを速度 50 %にする
    でなければ
```

画面上には半透明で表示されているよ

```
X を 右 のIRセンサの明るさ - 左 のIRセンサの明るさ にする
ラベル labelX に X を表示
L を 70 + X にする
R を 70 - X にする
ラベル labelL に L を表示
ラベル labelR に R を表示

ポート M1 のモータを速度 L %にする
ポート M2 のモータを速度 R %にする
```

この2つのプログラムを組み合わせると
方角が -90度より小さくも、90度より大きくもないときに
ボールを追いかけるプログラムに仕上げよう





サッカープログラム を完成させよう

- .01 試合のルールを確認しよう
- .02 ボタンで試合スタートできるようにしよう
- .03 試合にむけてもっと動きを良くしよう

試合のルールを確認しよう

S

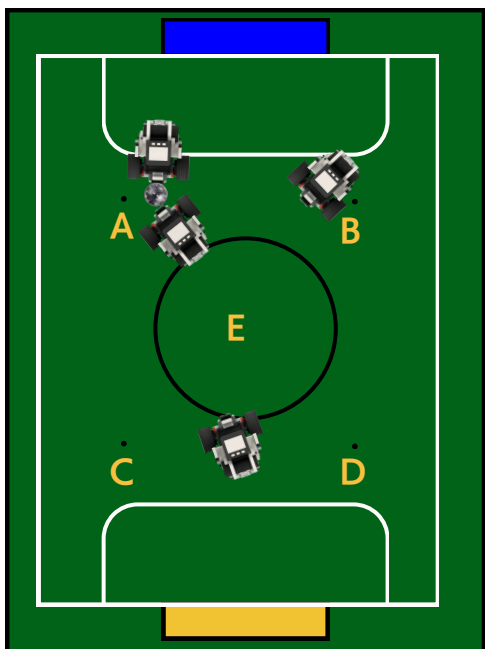
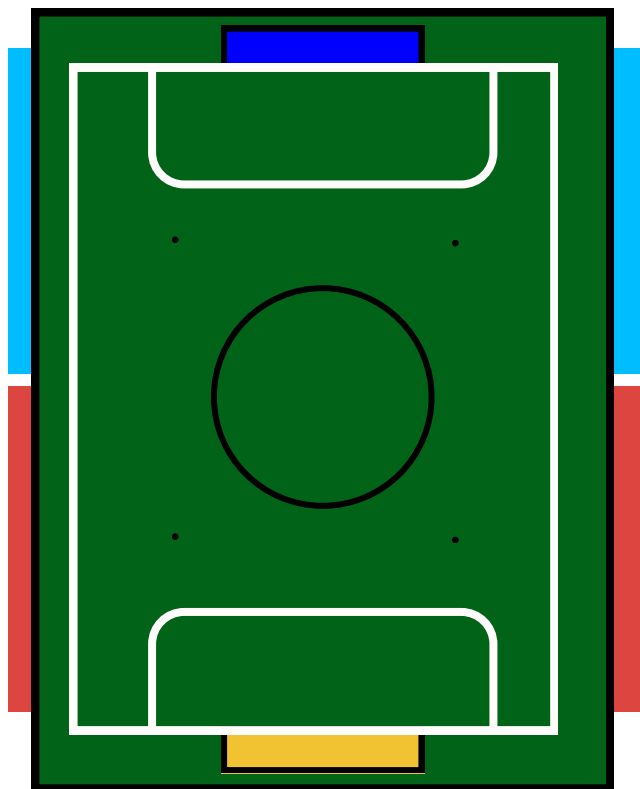
サッカープログラム
を完成させよう

キックオフ（試合開始）の手順
→先攻後攻はコイントスで決める

- ① ボールを中央に置く
- ② 先攻は **赤枠内** にロボットを置く
- ③ 後攻は **青枠内** にロボットを置く
(黒線の円に入ってはいけない)
- ④ コンパスセンサのボタンを押して敵陣を設定
- ⑤ **キックオフの合図にあわせてロボットを動かす**

試合中はロボットに触るのは禁止だよ。

- ・ 試合に戻るときは先生の指示に従って、ボールの反対側を向いてロボットを置く
- ・ ボールを全く追っていないロボットは故障として扱う



Q 試合が止まったとき、
ボールはどう移動されますか？

ロボットに近い「中立点」を順に移動します。
左の図の場合であれば A→B 最後は中心のEに移動。
それでも試合がすすまなければ「リスタート」します。

(リスタートは4台とも円の外から試合を再開します)

Q 試合が始まるとき、
2台とも自分のゴール前に置けますか？

ロボットに近い「中立点」を順に移動します。
左の図の場合であれば A→B 最後は中心のEに移動。
それでも試合がすすまなければ「リスタート」します。

置けません。ルールで禁止しています。
(リスタートは4台とも円の外から試合を再開します)

いまのプログラムのままだと
簡単に動きを開始したり・止めたりできないね。次のページでボタンを使って解決しよう！

ボタンで試合スタートできるようにしよう

S

サッカープログラム
を完成させよう

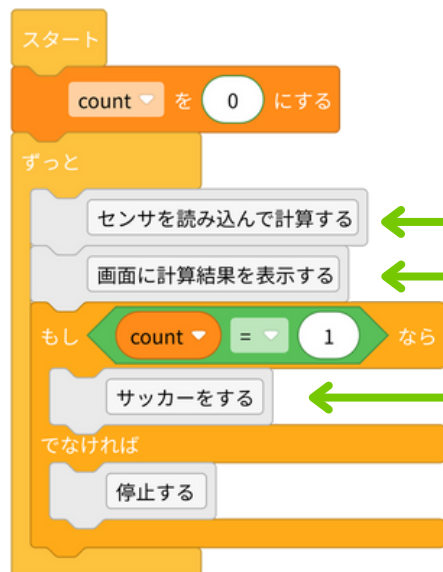
ボタンで 動きの開始・停止を操作できるように しよう。

変数 **count** をボタンが押された時に
変更することで、サッカーをするか
モータを止めるか条件分岐させよう。

QRを読み込んで
プログラミングを開始

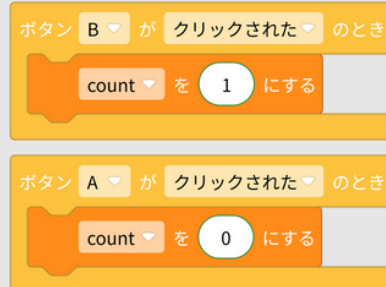


Soccer



サッカープログラムの中身を
役割ごとに切り分けて中に入れよう

ボタンが押されたときに
中のプログラムが実行されるよ。



試合にむけてもっと動きを良くしよう

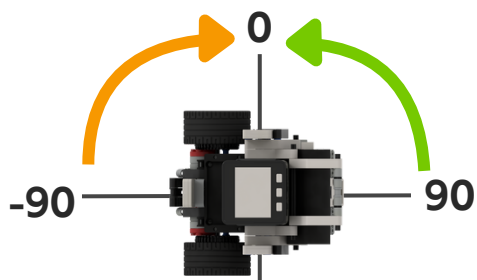
S

サッカープログラム
を完成させよう

真横を向いたとき 正面まで回転 しよう

Soccer

-90度以下や、90度以上を向いたとき正面を向くまで継続して回転するようにしよう。



QRを読み込んで
プログラミングを開始



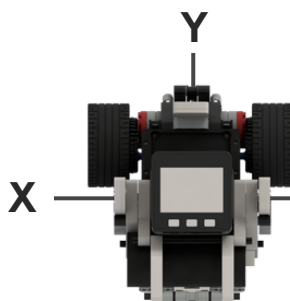
センサでつづける方法もあるけど
今回は  秒待つ を使おう。

テストのために別でプログラムを
作って速度と時間をはかるといいよ。

ボールの縦の位置を表す変数を作って 後ろに下がる

Soccer

ボールがロボットより後ろにあるときに下がれば、
敵ゴールを向いたままだもロボットの正面でボールを
捉えられるため得点しやすくなります。



ロボットとボールの縦の位置関係を変数Yで示そう。

X を  のIRセンサの明るさ -   のIRセンサの明るさ にする

ではボールの前後の位置を示す変数Yは
どこからどこのセンサの値を引けばよいでしょうか？

ボールがロボットの前にあるときYの値が大きくなるようにしてね。


Y を  のIRセンサの明るさ -  のIRセンサの明るさ にする

もし  >  なら

ポート M1 のモータを速度  % にする

ポート M2 のモータを速度  % にする

でなければ

ポート M1 のモータを速度  % にする

ポート M2 のモータを速度  % にする

自分で計算したYを使って
ボールがロボットよりも

- 前にあるのか
 - 後ろにあるのか
- によって、

- ボールを追いかける
 - バックする
- 動きを切り替えさせよう。

RoboCupJuniorって？ 自分たちが作ったロボットで世界を目指します。



ロボカップジュニアは、自分で作ったロボットを使って、世界中の仲間と競い合う大会です。

でも、それだけじゃありません。**RoboCupJunior** は、もうひとつの「居場所」
ノード大会▶ブロック大会▶全国大会▶世界大会 と少しずつ君の世界が広がっていきます。

大会の会場では年齢の上下関係なく、すごいロボットをみんなで見せてもらったり、君のロボットについて知りたい子が質問してきたり。**全国にロボット仲間ができます。**

ロボット製作を通じて身につく力

論理的思考

ロボットにはいろんな要素があり、それぞれが相互に働きあって動いています。
目標を細かく分け、条件分岐や繰り返しを組み合わせる手順を組み立てることで、筋道立てて考える力が身につきます。

課題解決力

つまずきも自分たちで乗り越える経験が、課題を解く力を伸ばします。
ロボット製作はハードとソフトの合わせ技。原因を切り分け、改良を重ねるプロセスを実践的に学べます。

協働力

チームでロボットを作るなかで、協力とコミュニケーション力が磨かれます。
大会ごとにロボットを紹介するプレゼンシートも作成するため、伝える力も伸びます。

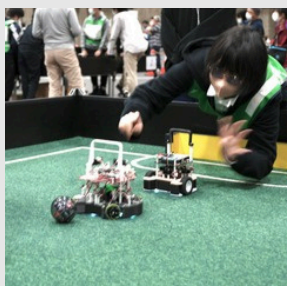
こんな競技があるよ

OnStageチャレンジについてはロボットラボでは実施しないため掲載しておりません。

サッカーチャレンジ

センサーを使ってボールを追いかけて、ゴールを決めよう。

動きの早さや正確性を追いかけて、ハード（機械）とプログラミングがうまくかみ合うことが勝利の鍵だよ。



レスキューチャレンジ

倒れた人を助けるレスキューロボットを作って、坂道・障害物などの難しいコースをクリアしよう。

ロボットの判断力と正確な動きが求められる競技だよ。





kusuwata

つくるものがヒトをつくる
創作意欲が羽ばたける世界をつくる



Website

ものづくりを通じ、自己有用感を生む「心のエンジン」を育てる教育の社会実装を目的に活動しています。

自己有用感とは「自分が誰かや社会の役に立ち、貢献できている」と実感する感覚で、存在価値への肯定感や行動の意欲を高める源になります。

Activity

● ロボット教室運営 (eとびあ・かがわ ロボットラボ)

● 公立小学校での実践授業

旧株式会社フレップテックより事業を承継し、
継続して公教育との関わりを模索しています。



授業のようす



総合学習 x テックの実験授業



国語 x ロボットの実験授業



コピキタス研究会への参加

● ロボット教材開発

M5Stackという組み込み向けデバイス
の活用・改造から、ビジュアルベースの
プログラムツールの専用カスタマイズ・機能
開発を行っています。



5年を超える指導実績から導き出された
実際の教室での運用に則した教材の開発を
行っています。GoogleDriveと連携できる
独自の教室用ファイルマネージャも開発し、
円滑な授業実施に貢献します。

Member



楠田 亘

Kusuda Wataru

ロボットラボ出身

kusuwata 代表

- 2011 ロボカップジュニア全国大会6位
～小・中学校にて不登校を経験～
- 2016 香川高等専門学校入学
- 2016 株式会社フレップテック創業
- 2018 かがわビジネスチャレンジコンペ 最優秀
- 2019 eとびあかがわロボットラボ受託開始
- 2020 資金難で事業継続を断念
- 2021 経営塾「大坂塾」で業務に従事
- 2023 kusuwataとして活動再開



天竺 寛貴

Tenjiku Hirotaka

ロボットラボ出身

hrecode 開発

- 2020 香川高専詫間キャンパス 情報工学科 入学
- 2022 CyberSakura 2位
- 2022 情報危機管理コンテスト アグレッシブ賞
- 2022 高専プロコン 課題部門「Voice Cat」 優秀賞
- 2023 高専プロコン 自由部門「わんもあ」 最優秀賞
- 2024 高専プロコン 自由部門「Uni」 最優秀賞
- 2025 香川大学 創造工学部 入学

「サッカーロボットをつくろう」テキスト開発参加メンバー



久保 美羽
Kubo Mihane

ロボットラボ出身



森 悠造
Mori Yuzo

ロボットラボ出身



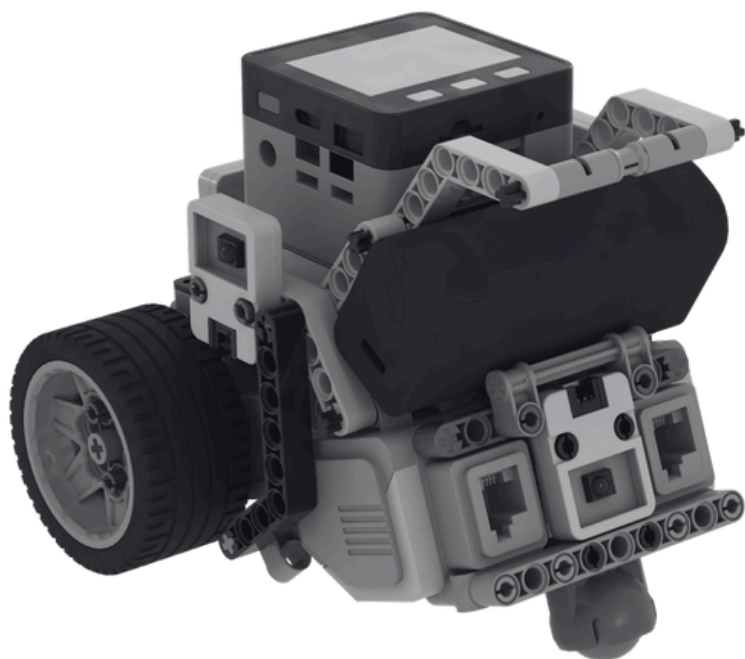
日野 正康
Hino Seiko



岩部 夏季
Iwabu Natsuki



太田 祐希
Ota Yuki



つくることがヒトをつくる
創作意欲が羽ばたける世界をつくる

kusuwata