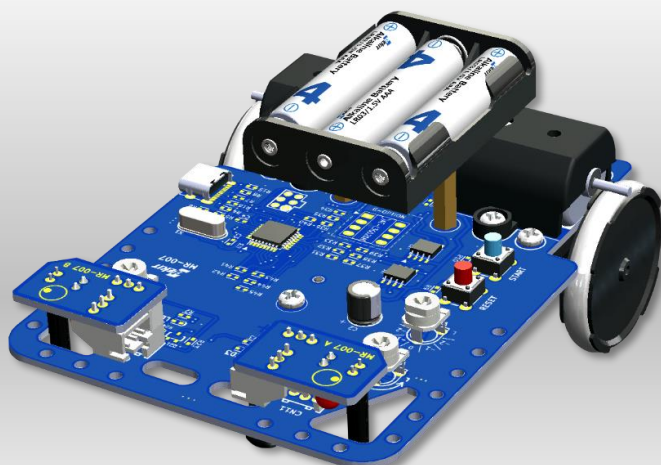


# コロボライト2 MR-007 プログラミング・ガイド



## ようこそ、ロボットプログラミングの世界へ

みなさんの周りにはどんな働くロボットがありますか？

ロボット掃除機が家の掃除をしたり、自動運転の車や、人工知能を搭載したロボットが将棋でプロ棋士と対決したりと、今や生活の様々な場面でロボットを見ることができます。

このように活躍しているロボットは「プログラミング」をすることで人が操縦することなく自律的に動くことができます。

みなさんもコロボライト2をプログラミングして、ロボットをプログラムで動かしながら、プログラムの基本をまなんでいきましょう。


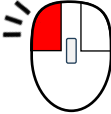

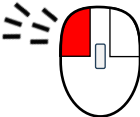
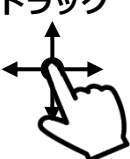
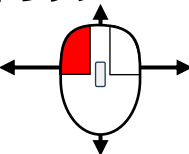
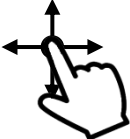
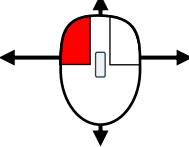

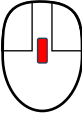

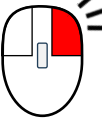


このガイドの中では、コロボライト2のことをロボットと記していきます。



# タッチパネル操作とマウス操作

このガイドで書いているタッチパネルでの操作に対応するマウス操作は以下のようになります。お使いのパソコンの環境に合わせて操作してください。

タッチパネル操作		同じ動作をするときのマウス操作	
<b>タップ</b> 	画面上のアイコンやワークスペースに指1本で軽く触れ、指を離します。	<b>クリック</b> 	マウスの左ボタンを押して指を離します。
<b>ダブルタップ</b> 	画面上のアイコンを指で2回連続でタップします。	<b>ダブルクリック</b> 	画面上のアイコンをマウスの左ボタンで2回連続クリックします。
<b>ドラッグ</b> 	画面上のアイコンに触れ、指を離さずに目的の場所まで動かして指を離します。	<b>ドラッグ</b> 	画面上のアイコンをクリックしてボタンを押したまま、目的の場所まで動かして指を離します。
<b>スクロール</b> 	画面のドックやワークスペースに触れ、指を払うように動かします。	<b>スクロール</b> 	画面上のドックやワークスペースをクリックしたままマウスを動かします。
<b>拡大・縮小</b> 	拡大の場合は、指1本で画面に触れたまま、別のもう1本の指で画面に触れ、2本の指先を開くように動かします。 縮小する場合は、逆に2本の指先でつまむように動かします。	<b>ホイール</b> 	マウスのホイールを回します。
<b>長押し</b> 	画面上のアイコンを指1本で、2秒以上触れたままにします。	<b>右クリック</b> 	マウスの右ボタンを押して指を離します。

# 1. プログラミング・ポータル



## プログラミングの準備



ブラウザには Google Chrome または Microsoft Edge を使います。



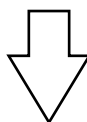
1

## コロボライト2 プログラミング・ポータル にアクセスする

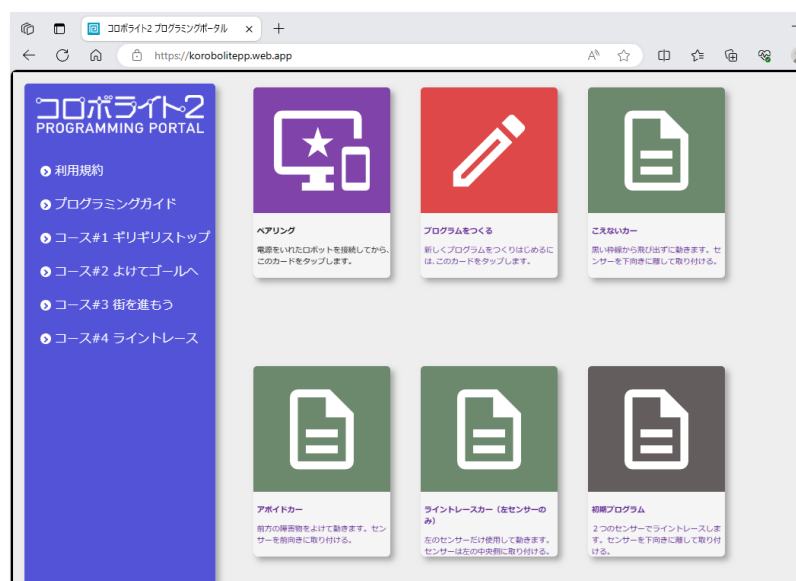
ブラウザのアドレスバーに下記URLを入力してサイトに移動します。

**korobolitepp.web.app**

※完全なURLは「https://korobolitepp.web.app」ですが、多くの場合、上記のように省略された形で入力できます。



コロボライト2 プログラミング・ポータルのページが開きます。





# プログラミングの準備

## 2 ペアリングする

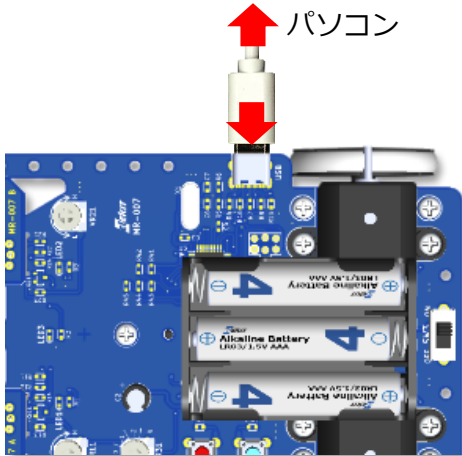


ロボットを始めてパソコンにつないだときは「ペアリング」をします。

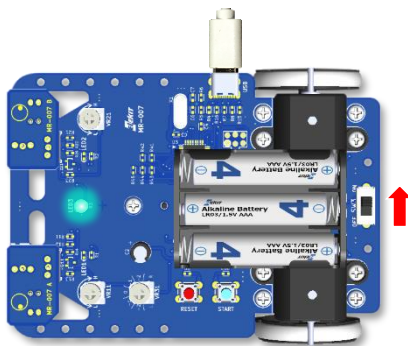
ペアリングを行っていない状態では、**ロボットとWebアプリとの通信ができません。**

※使用するブラウザが変わったときや、ブラウザを使用するログインユーザーが変わったときにもペアリングが必要なことがあります。

- ① ロボットとパソコンをUSBケーブルで接続する



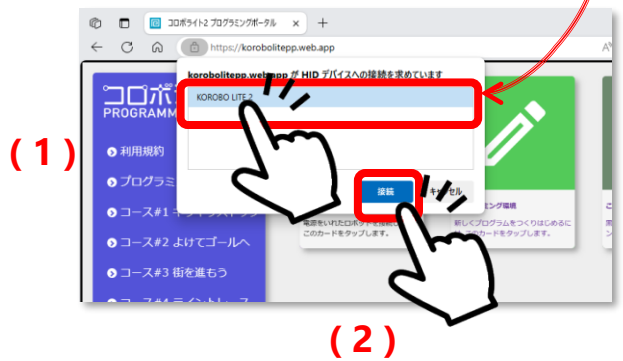
- ② 電源スイッチをONにする



- ③ 「ペアリング」をタップ



- ④ (1) **KOROBO LITE 2** をタップして選ぶ  
(2) **接続** を押す



- ⑤ **OK** をタップ

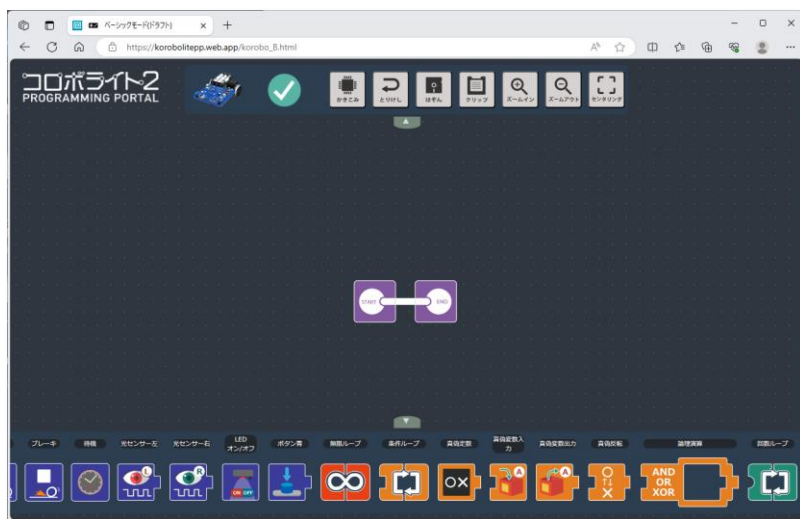
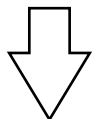
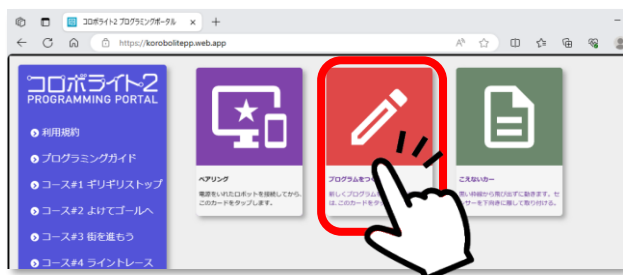


- ⑥ ペアリング終了

ペアリングは一度だけ行えばよく、次回以降は、ペアリングは必要ありません。

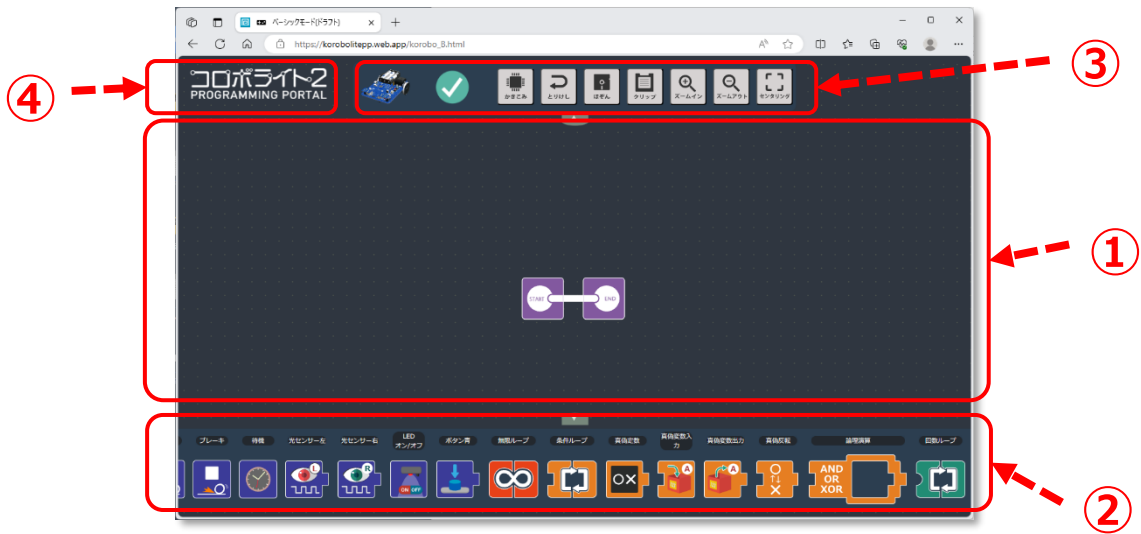
## 3 プログラム編集画面を開く

プログラムを新しく作るときは、プログラミング・ポータル画面の、プログラムをつくるカードをタップします。



プログラム編集画面が開きます。  
さあ、プログラミングをはじめよう！

## プログラム編集画面



### ①ワークスペース

この場所にプログラムを作成します。

スタートとエンドのアイコンはあらかじめ配置されています。

### ②ドック

プログラムで使用する命令アイコンが並んでいます。前進や後進などの移動するためのアイコンや、センサーの情報で分岐やループするアイコン、数値や回数を設定するアイコンなどが用意されています。



### ③スイッチボード

ロボットの接続状態を表示したり、プログラムの保存や、プログラムの書き込みなどの操作をするためのアイコンが並んでいます。

### ④プログラミング・ポータルへ



このロゴをタップすると、「プログラミング・ポータル」ページに移動します。

## 📖 スイッチボードのアイコン



### USB チェック

ロボットと正しく接続されているときに緑色のチェックマークが表示されます



### かきこみ

プログラムをロボットに書き込み(転送)します



### とりけし

プログラム編集操作を1つ前の状態に戻します



### ほぞん

作成したプログラムを保存します



### クリップ

現在プログラムを表示しているブラウザ画面のURLアドレスを、クリップボードに保存します。このURLをメールやメモ帳に貼り付けて他の人に伝えることで、別のパソコンのブラウザで同じ画面が開きます。



### ズームイン ズームアウト

ワークスペース表示の拡大、縮小をします



### センタリング

ワークスペースをセンタリングします

## 📖 ドックのアイコン

ドックのアイコンの詳しい説明は、「8. アイコンの説明」で説明しています。次ページ以降でプログラミングに慣れて、オリジナルのプログラムを作るときに活用してください。



# プログラミング・ポータル画面の説明

## プログラミング・ポータル画面



### ①「ペアリング」カード

ロボットをペアリングします。

### ②「プログラムをつくる」カード

あたらしくプログラムをつくるときにタップします。

### ③「プログラムカード」

作成済みのプログラムがカードとして表示されます。



## プログラムを削除する

削除したいプログラムのカード上で長押しして指を離すとメッセージが表示されるので、OKを押します。

※最初から収録されているプログラムを消すことはできません。

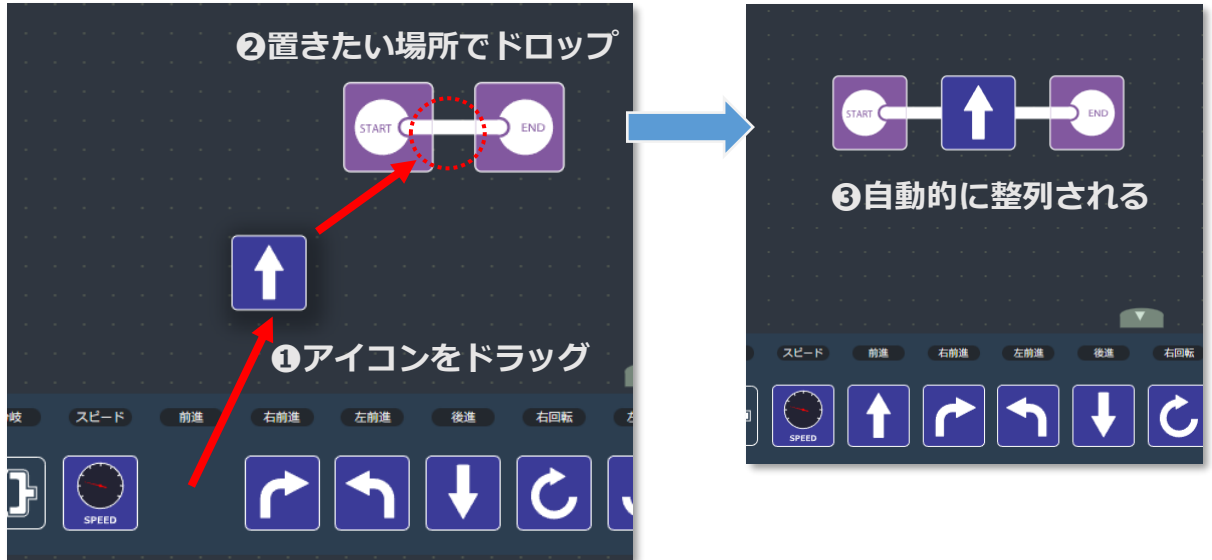






# アイコンの操作

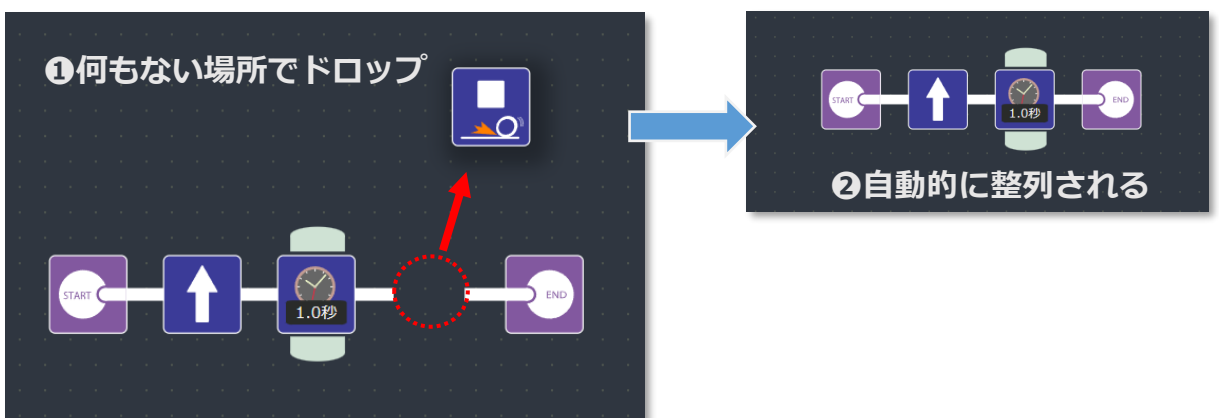
## ①アイコンを置く



## ②アイコンをならびかえる




## ③アイコンを消す





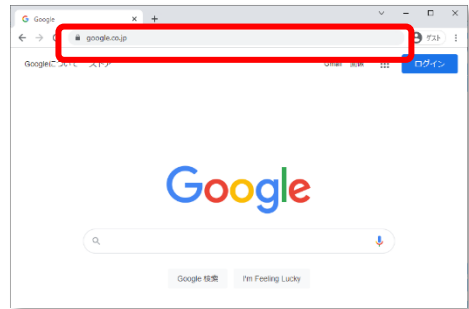
# プログラムのコピー

プログラムを作ってうまく動作したときは、プログラムのファイルを保存しておくの良いかもしれません。試行錯誤しているときに「昨日のプログラムに戻したいけど上書してしまって戻れない…」となってしまう前に保存しておきましょう。

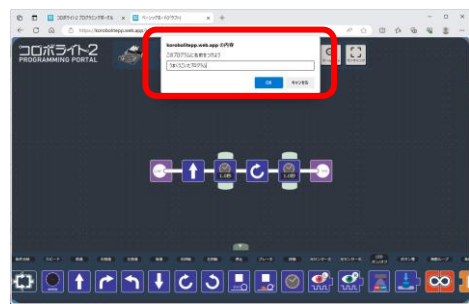
- ① 保存したいプログラムを開き、  
クリップ  をタップします。  
(URLがコピーされます)



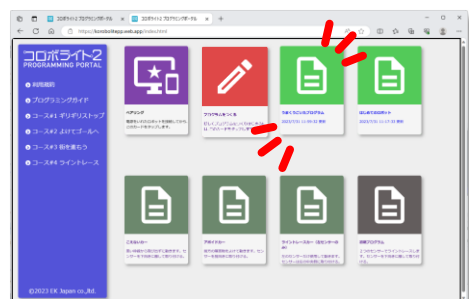
- ② ブラウザーの新規ページを開き、  
アドレスバーにURLを張り付けて、  
移動します。



- ③ プログラムの画面が開いたら、  
ほぞん  をタップし、  
名前をつけてOKをタップします。



- ④ プログラムが保存されます。



## 2. プログラムのかきこみと実行



### この章で使うアイコン



#### スタート

このアイコンからプログラムが始まります。



#### 前進

ロボットが前へ向かって進みます。



#### エンド

全ての動作を停止し、プログラムを終了します。



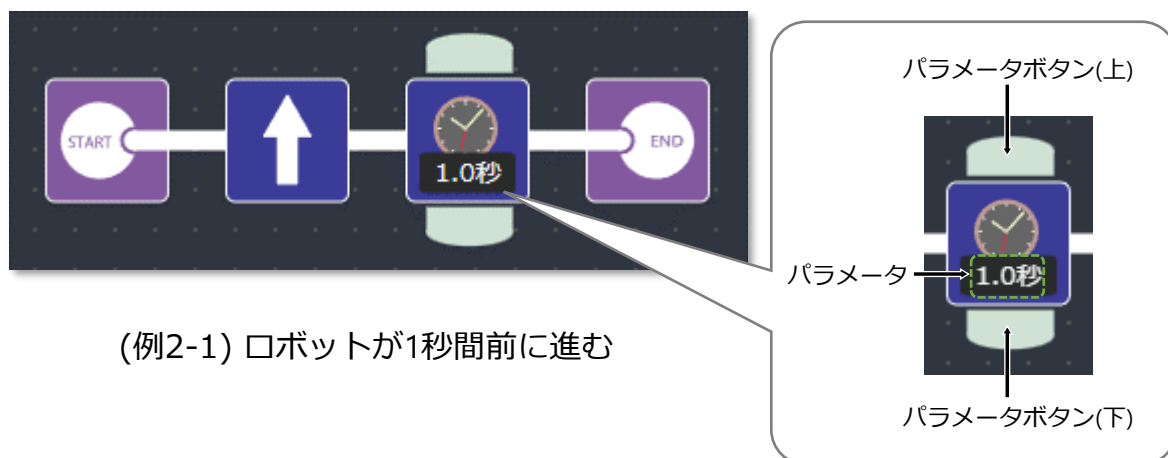
#### 待機

指定された秒数だけプログラムを一時停止します。



### 前に進むプログラムをつくる

下図のような「ロボットが1秒間前に進む」プログラムをつくります。待機アイコンの秒数は1秒に設定しましょう。アイコンの上下についているパラメータボタンをタップすると秒数が変わります。



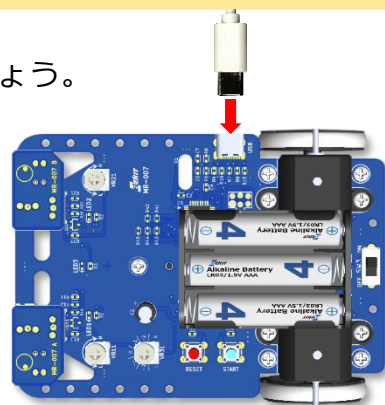


## プログラムの書き込み

プログラムができたなら、プログラムを書き込みましょう。

- ①ロボットとパソコンをUSBケーブルで  
接続します。

付属のUSBケーブルを使います。

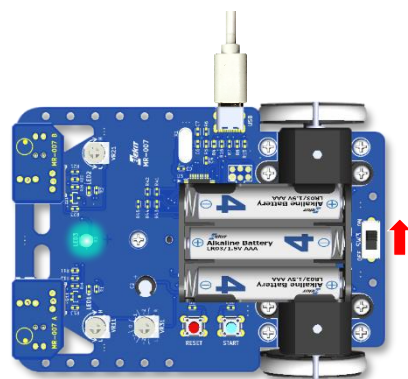


- ②ロボットの電源スイッチをONにします。

電源スイッチをONにするとLED3(緑)  
が点灯することを確認します。

LED3が点灯しないときは、

- ・ロボットのリセットボタン(赤)を一回押してみます。
- ・USBケーブルを一旦抜いてもう一度差し込んでみます。




- ③かきこみボタンをタップしてプログラムを書き込みます。

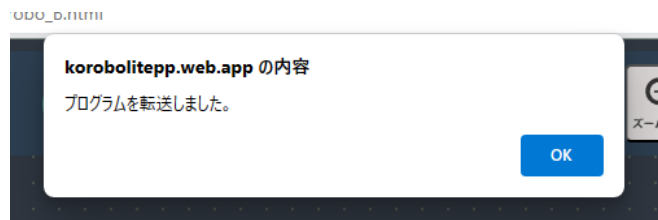



チェックマーク  が表示されていると  
きにプログラムを書き込むことができます。



未接続マーク  が表示されているとき  
はプログラムを書き込むことができません。  
ケーブルの接続の確認やロボットの電源ス  
イッチの位置を確認します。

- ④画面に**プログラムを転送しました。**と表示されれば書き込み完了です。

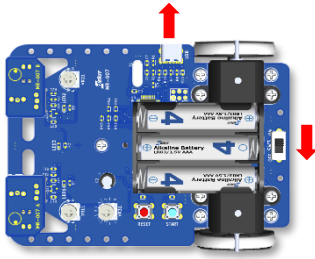


 ボタンをタップして  
メッセージを消します。

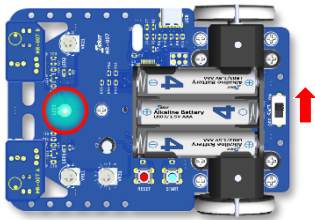
★この表示が出るまで、ケーブルを抜かないこと!

## ☑ プログラムの実行

書き込んだプログラムを実行してみましょう。スタートスイッチを押すと「1秒前進する」という動作をします。うまく動いたら成功です！



電源をOFFにして、  
USBケーブルを抜きます



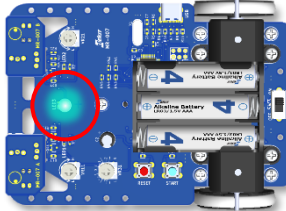
電源をONにすると  
LED3が数回点滅します



スタートボタン(青)を  
押すと、LED3が数回点滅した後  
プログラムを実行します



1 秒間前進！



プログラムが終了すると  
LED3がゆっくり点滅します。



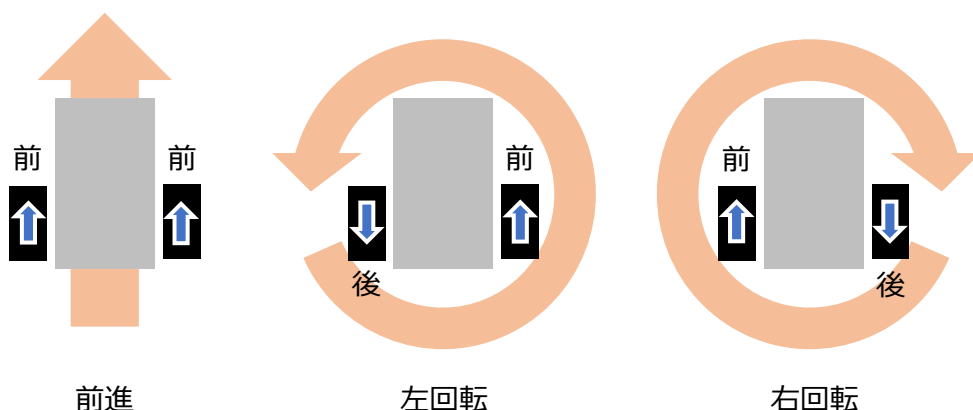
もう一度プログラムを実行するときは、  
リセットボタン(赤)を一回押してからス  
タートボタン(青)を押します。

# 3. いろいろな移動をやってみよう

## 📖 ロボットの動きをよく観察してみよう

ロボットは左右に1つずつ車輪がついています。前進させるプログラムを実行して動いているときに車輪が回転している方向はどのようになっていたでしょうか？車輪は右も左も、鉄棒で前回りをするのと同じ前向きに回転していたと思います。

ここではロボットに後進や回転をさせてみます。ロボットが動いているときに左と右の車輪が前向きに回転しているか、後ろ向きに回転しているか、よく観察してみましょう。



## この章であたらしく使うアイコン



### 後進

ロボットが後ろへ向かって進みます。



### 左回転

ロボットがその場で左へ回転します。



### 左前進

左斜め前に曲がりながら進みます。



### 右回転

ロボットがその場で右へ回転します。



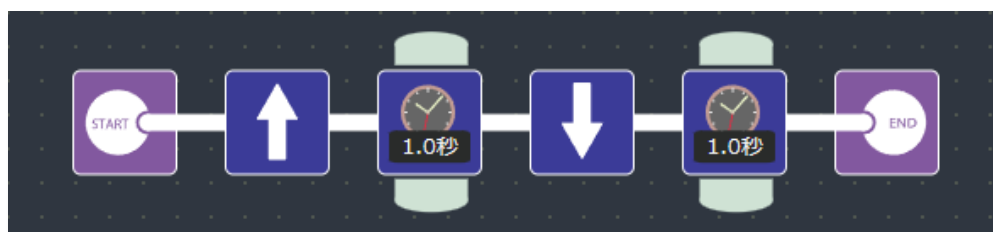
### 右前進

右斜め前に曲がりながら進みます。



## 1秒間前に進む→1秒間後ろに進む

下図のような「1秒間前進→1秒間後進」するプログラムを作ります。待機アイコンのパラメータは1.0秒に設定しましょう。



(例3-1) 1秒間前進→1秒間後進



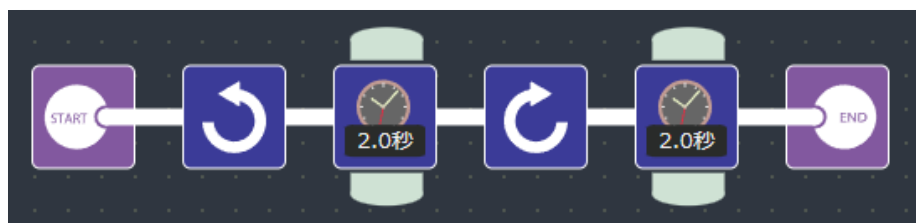
### 動作チェックをしよう

- START → 前進(1秒間) → 後進(1秒間) → END  
の動きはうまくできましたか？

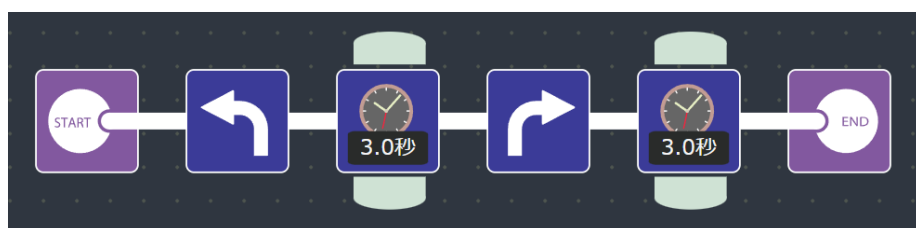


### やってみよう

「左回転」と「右回転」、「左前進」と「右前進」を使って、下図のようなプログラムを作ります。どんな動きをするか予想して、実際に動かしてみましよう。ロボットが動いているときに車輪がどちらの方向に回転しているか、よく観察してみてください。



(例3-2) 左回転、右回転を使ったプログラム



(例3-3) 左前進、右前進を使ったプログラム



## パラメータの便利な入力方法

待機アイコンは0秒～10秒の範囲で時間を設定できます。  
パラメータボタンをタップして値を設定することもできますが、  
アイコンのパラメータを短くタップすると、入力ウィンドウが表示され、  
キーボードで値を入力することもできます。





## 4. 同じ動きを繰り返そう



### 同じ動きのアイコンをたくさん並べる？

例えば、(例3-1)で作った「1秒間前進→1秒間後進」を5回繰り返すプログラムを作ろうとしたとき、どうすれば実現できるでしょうか？



「1秒間前進→1秒間後進」を5回繰り返す

これで動かすことができます。でもこのプログラム、前進と後進を繰り返すだけの単純な動きなのにとっても長くて読みにくいですね。それに「繰り返す回数を100回にしてください」なんて言われたらもう大変ですね。

この章ではこういった単純な動作を繰り返し行うときに便利な「繰り返し処理」について学んでいきます。



### この章であたらしく使うアイコン



#### 回数ループ

囲まれた部分のプログラムを数値定数で指定した回数だけ繰り返します。



#### 数値定数

回数ループなどに使う数値を指定します。  
パラメータボタンによって数値を増減できます。



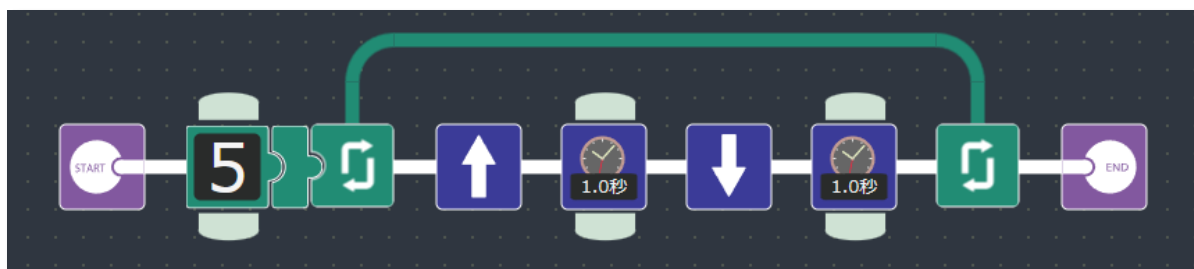
#### 無限ループ

囲まれた部分のプログラムを無限に繰り返します。



## 「1秒間前進→1秒間後進」を5回繰り返す

「1秒間前進→1秒間後進」を5回繰り返すプログラムをつくります。繰り返す動作を「回数ループ」のはじめと終わりのアイコンの間に置き、繰り返す回数を「数値定数」で設定します。数値定数アイコンのパラメータボタンを数回クリックして値を5にします。



(例4-1) 「1秒間前進→1秒間後進」を5回繰り返す



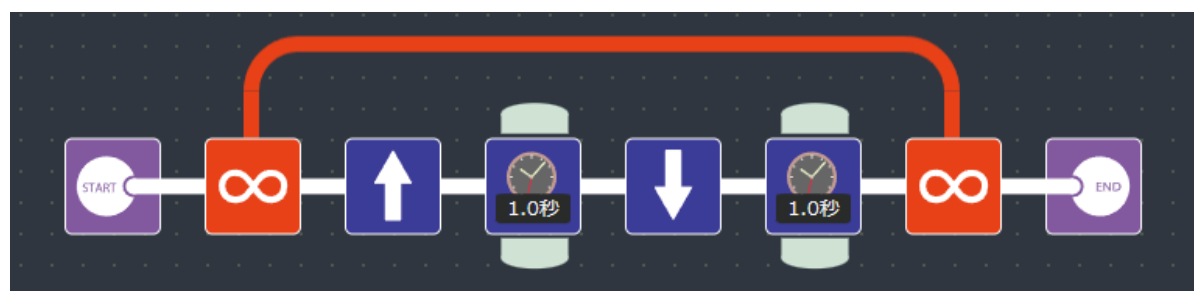
### 動作チェックをしよう

- 「前進(1秒間)→後進(1秒間)」を5回繰り返して動きますか？



### やってみよう

自動でロボットを動かす場合、プログラムが終了することなく、無限に動き続けてほしいことがほとんどです。「無限ループ」を使って、左回転と右回転をずっと繰り返すプログラムを作ってみましょう。

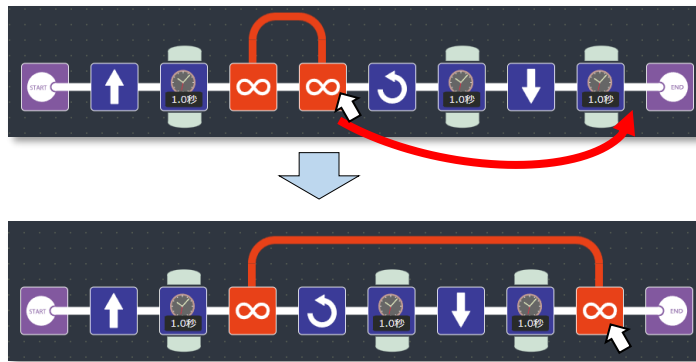


(例4-2) 無限ループ

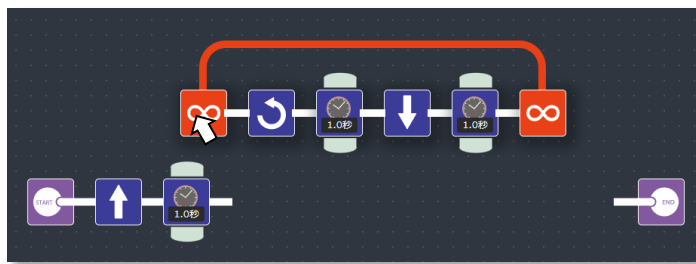


## 無限ループ/回数ループの操作方法

ループアイコンの後ろ側のアイコンをドラッグして囲む範囲を変えることができます。



ループアイコンの先頭側をドラッグするとループアイコンで囲まれている範囲を一緒に移動することができます。



## パラメーターの便利な入力方法

数値設定アイコンは-100~100の範囲で数を設定できます。パラメータボタンをタップして値を設定することもできますが、アイコンのパラメータを短くタップすると、入力ウィンドウが表示され、キーボードで値を入力することもできます。





## 数値を受け取るアイコン

「回数ループ」アイコンは、直前においたアイコンから数値を受け取るの  
です。直前においたアイコンは「数値をわたす」ものでなくてははいけ  
ません。プログラミングにつかえるアイコンのうち、「数値をわたす」も  
の、**「数値定数」**、「**数値変数出力**」などがありますが、アイコンのかた  
ちで見分けることができます。



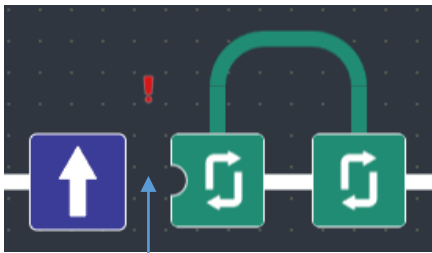
「丸いでっぱり」は、  
次のアイコンに「数  
値」をわたすことを  
表す

「丸いへこみ」は、  
直前のアイコンから  
「数値」を受けとるこ  
とを表す



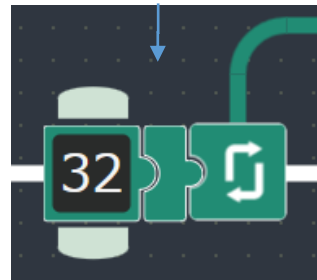
### 正しい組み合わせ

「数値定数」アイコンが「32」という数  
値を次にわたし、「回数ループ」アイ  
コンがそれをうけとっている



### 正しくない組み合わせ

「回数ループ」アイコンに数値がわた  
されていない。そのため、「まちが  
い」をあらわす「！」が表示されている。



## 5. スピードをかえてみよう

### 📖 スピードの役割

人間は道をゆっくり歩いているときに、目の前に急にボールがとびだしてきたら、すぐに止まることができますよね。でも全力で走っているときだとどうでしょうか。きっとすぐには止まれないと思います。ロボットも同じで、早いスピードで動いていると、センサーが反応してもすぐには止まらず、少し進みすぎる場合があります。センサーでモノを感知するときは、ロボットのスピードを調整したほうがよい場合もあります。ここではスピードを変えるプログラムを作ってみましょう。



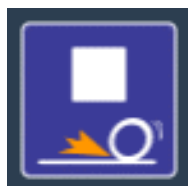
### この章であたらしく使うアイコン



#### スピード

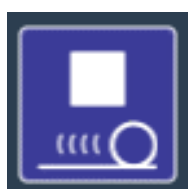


ロボットのモーターの回転するスピードが速い[FAST]か、遅い[SLOW]、その間の[MID]のどのスピードで回転させるか設定します。このアイコンを置いた後は次に設定するまで設定したスピードになります。  
※スピードアイコンを全く置かない場合のスピードはFASTになります。



#### ブレーキ

ロボットのモーターに電氣的なブレーキを掛けて回転を停止します。



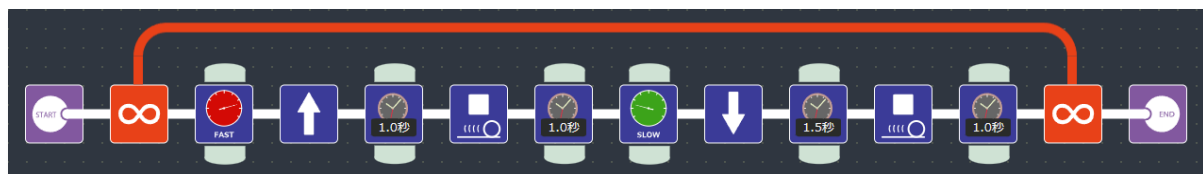
#### 停止

ロボットのモーターの動作をOFFにします。



## 「1秒早く前進 → 1.5秒遅く後進」をずっと繰り返す

「1秒間早く前進→1.5秒間遅く後進」をずっと繰り返すプログラムをつくりま  
す。前進と後進が切り替わる時にロボットを一旦停止させることもやってみ  
ましょう。



(例5-1) 「1秒間早く前進→1.5秒間遅く後進」を5回繰り返す



### 動作チェックをしよう

- 「早く前進(1秒間)→ゆっくり後進(1.5秒間)」を繰り返して動きますか？



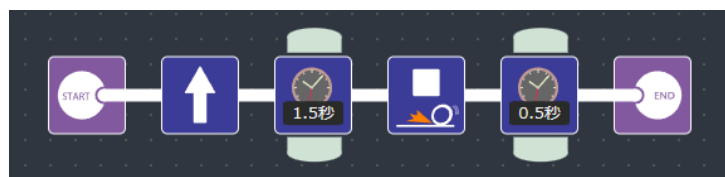
スピードを[SLOW]にしたときにロボットの動きが途中で  
止まるときは電池が消耗している場合があるので、そんな  
症状のときは電池をチェックしてみましょう。



### やってみよう

ロボットを止める方法は2つあります。「ブレーキ」と「停止」です。  
ブレーキはモーターに電氣的にブレーキをかけるので、ピタッと止まりますが、  
停止はモーターを回すことをやめるだけなので、ほんの少しだけ進んで止まり  
ます。ブレーキと停止をプログラムして動きの違いを確認してみましょう。

ブレーキ



(例5-2)

停止



(例5-3)

## 6・センサーを使ってみよう

### 📖 センサーの役割

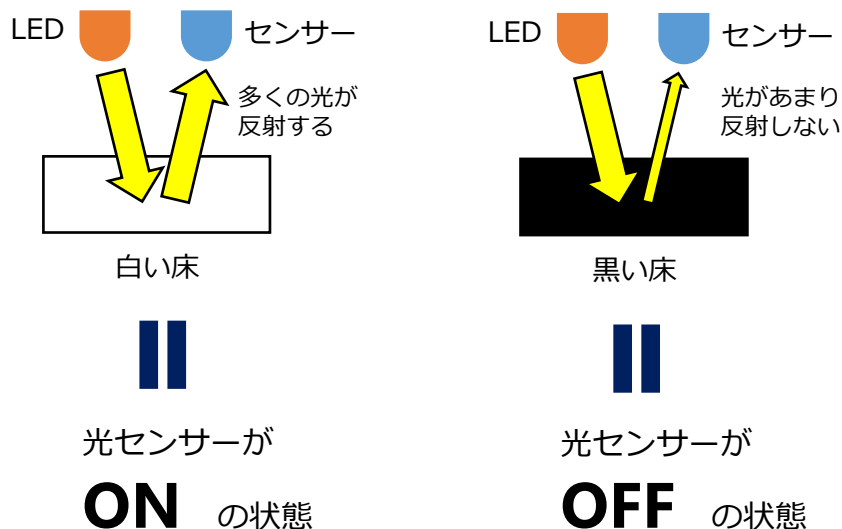
人間は目で見たり、耳で音を聞いたりしてまわりの様子を感じますが、世の中にあるロボットは目や耳のかわりにさまざまな種類のセンサーを使います。センサーには、壁にぶつかったことを感知するタッチセンサー、リモコン信号を感知する赤外線センサー、揺れを感じる振動センサーなど、いろいろなセンサーがあります。

コロボライト2には光センサーが使われています。ここでは光センサーを使ったプログラムを作ってみましょう。

### 📖 光センサーのしくみ

光センサーは受光部（光の明るさを見る目のような部分）に入ってくる光が明るい暗いを見ることができます。

コロボライト2の光センサーには、受光部（センサー）と発光部（LED）があります。LEDから出てくる光が壁や床に反射したときの光をセンサーが感知することで、床の色の明るさなどを見ることができます。

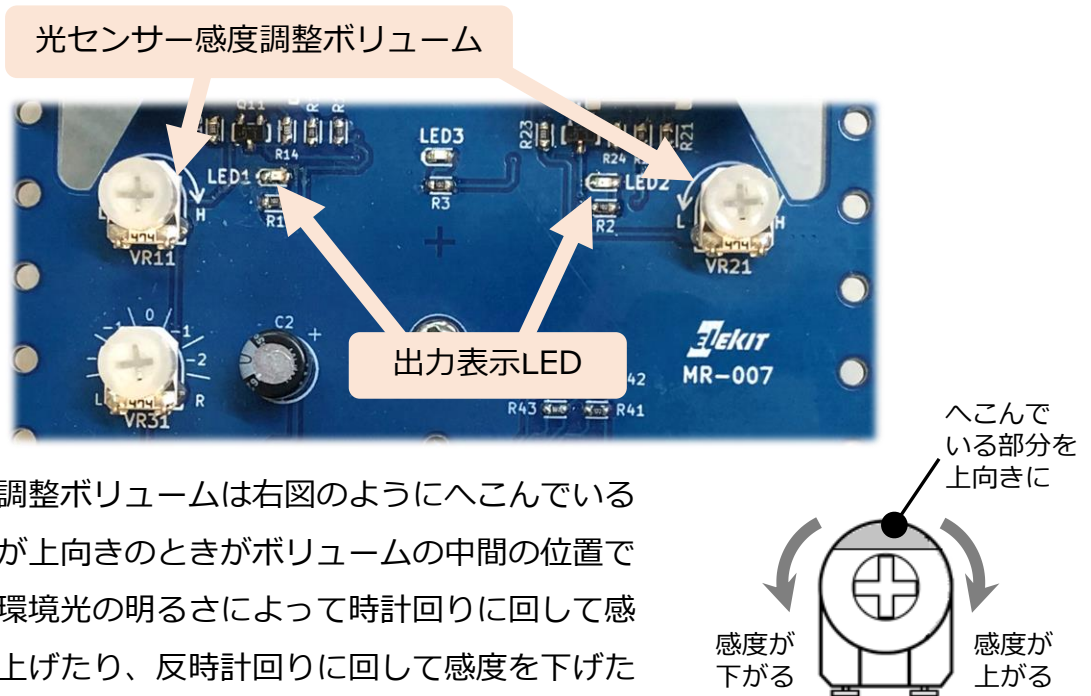




## 場所によって感度調整をしよう

光センサーは受光部に入ってくる光の明るさによって床の白・黒などを判別するため、環境光（室内の電灯や太陽光などの光）の影響を受けます。環境光が明るすぎるときは黒い床を検出しにくくなったり、環境光が暗すぎるときは白い床を検出しにくくなります。そのため、光センサーの下の床をLEDで照らして、環境光の影響をできるだけ小さくします。

さらに、確実にラインを検出できるようにメイン基板のVR11, VR21を使ってセンサーの感度調整を行いましょう。



感度調整ボリュームは右図のようにへこんでいる部分が上向きなときがボリュームの中間の位置です。環境光の明るさによって時計回りに回して感度を上げたり、反時計回りに回して感度を下げたりすることが必要になる場合もあります。



**光センサーの下が黒いときは、出力表示LEDが消えていて、光センサーの下が白いときは、出力表示LEDが点灯するように調整します。**

部屋の中に太陽光が入ってきているなど、環境光が明るすぎる場合には、ボリュームによる感度調整をしてもうまく動かないときもあります。その場合は窓のカーテンを閉めたり、電灯を暗くしてみたりして環境光が影響しないような工夫をしましょう。





## この章であたらしく使うアイコン



### 条件分岐

センサーが反応していたら分岐の○側、反応していなかったら分岐の×側のプログラムを実行します。



### 光センサー右

メイン基板の右側コネクタ(CN21)に接続した光センサーの状態を見るときに使います。



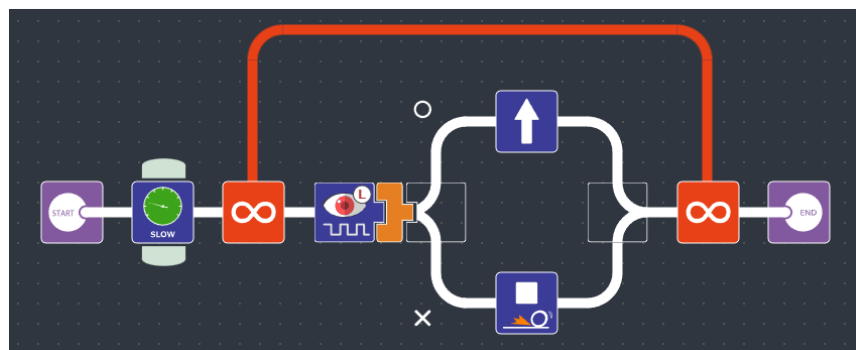
### 光センサー左

メイン基板の左側コネクタ(CN11)に接続した光センサーの状態を見るときに使います。



## 6 - 1 光センサーがOFFになったら停止する

条件分岐を使って、左側に取り付けた光センサーがOFFになったら停止するプログラムをつくります。センサーがONのときは前進しましょう。ロボットのスピードはSLOWにしておきます。

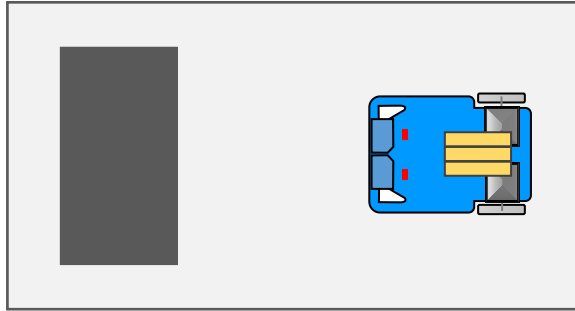


(例6-1) 左側の光センサーがOFFになったら停止



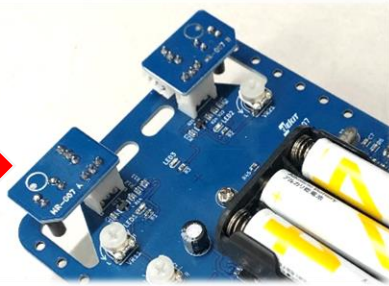
## 準備しよう

- ① ロボットを走らせるためのフィールドを準備しましょう。  
大きめの白い紙や床に、黒い紙または幅が広い黒テープなどで、下の図のようなフィールドをつくります。



- ② ロボットの左側のコネクタに光センサー基板Aが下を向くように差し込まれていることを確認しておきます。

光センサー基板Aを  
左側に差し込む



## 動作チェックをしよう

- 白い紙の上では前進して、光センサー左が黒い紙の上にくると停止しますか？



## やってみよう

- 光センサー左のアイコンを光センサー右に変えたプログラムを作ってみて、光センサー右で停止するようにロボットの動きを変えてみましょう。
- ロボットのスピードをFASTに変えてみて、ロボットが停止する場所がどうなるか観察してみましょう。
- 黒い線の幅を細くしてみて、ロボットの動きがどう変わるか観察してみましょう。

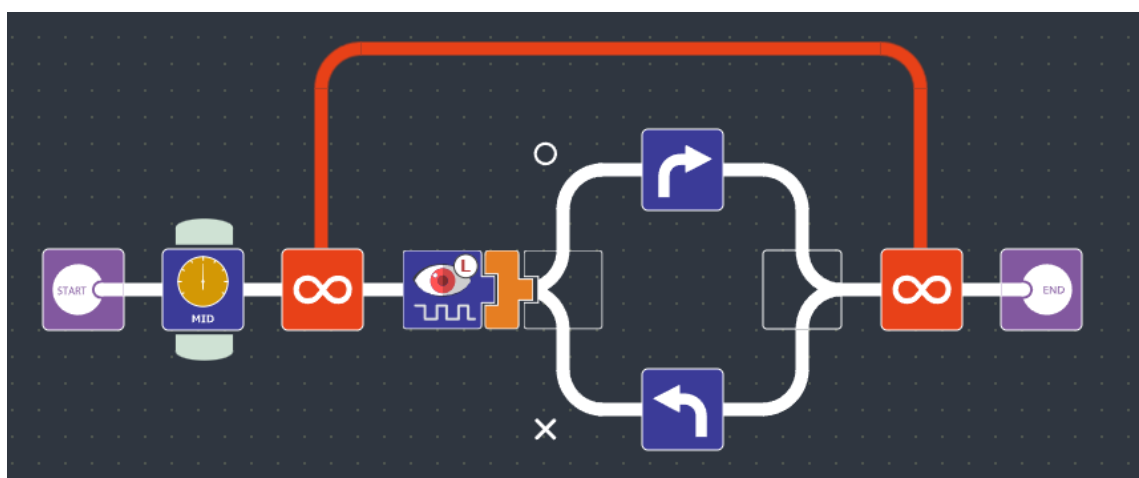


## 6-2 光センサーを使ってラインレースをする



### ラインレースする その1 左だけ使う

光センサー左だけを使って、床の白と黒を判別し、ラインに沿って進むプログラムを作成します。



(例6-2) 光センサー左だけでラインレースをする



### プログラムを観察してみよう

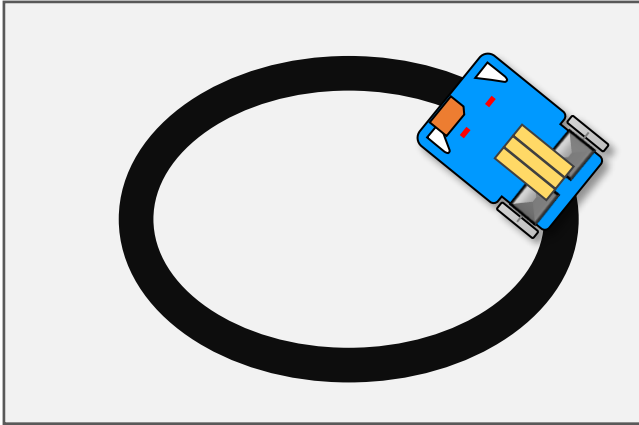
光センサー左だけを使っています。

光センサーは床の色が白ければ光センサーの状態は○になり、床が黒ければ×になります。○のときには右前進、×のときは左前進させています。ロボットのスピードが速すぎるとラインを上手く見つけることができず、コースアウトするかもしれませんので、スピードをMID(中間)にしています。



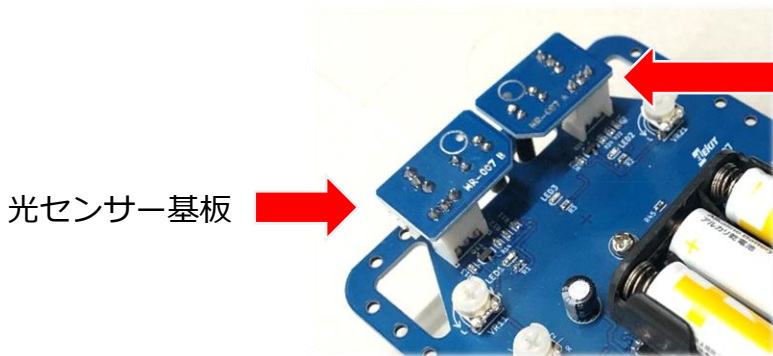
## 準備しよう

- ① ロボットを走らせるためのコースを準備しましょう。大きめの白い紙や床に、幅が広い黒テープなどで、下の図のようなコースをつくります。線の幅はできるだけ広い方が良く、2cm以上できれば3cm程度の幅を目安にします。



黒い線を作るときは、急カーブにならないようにつくります。  
急カーブがあるとそこでコースから外れやすくなります。

- ② ロボットの左側のコネクタに光センサー基板が下を向くように、センサーがロボットの中心に近くなるように差し込みましょう。



光センサー基板  
こちらは付けなく  
てもOK



## 動作チェックをしよう

- 黒いラインの上をロボットが進み続けていますか？



## 動きを観察してみよう

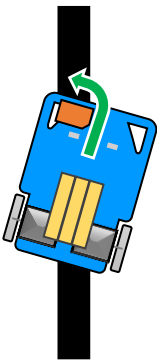
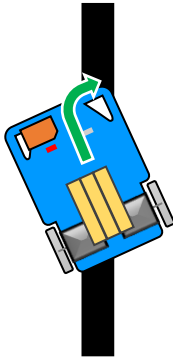
ロボットが光センサーでどのようにして黒いラインを検知してうまくラインをたどっているか、ロボットの動きを表で整理してみましょう。

プログラム上で光センサーの条件分岐の○と×は、

○：光センサーが白を検知しているとき

×：光センサーが黒を検知しているとき

を意味します。

ライン（白・黒） の状態	光センサ左 (L)の状態	光センサ右 (R)の状態	ロボットの動き
	× (黒)	— (チェックしない)	左センサーの下にラインがあるので <b>左前進</b>
	○ (白)	— (チェックしない)	ラインが右にあるので <b>右前進</b>

このようにして適切な動きをプログラミングすることで、ロボットは床の色（白・黒）を判別してライントレースをすることができます。

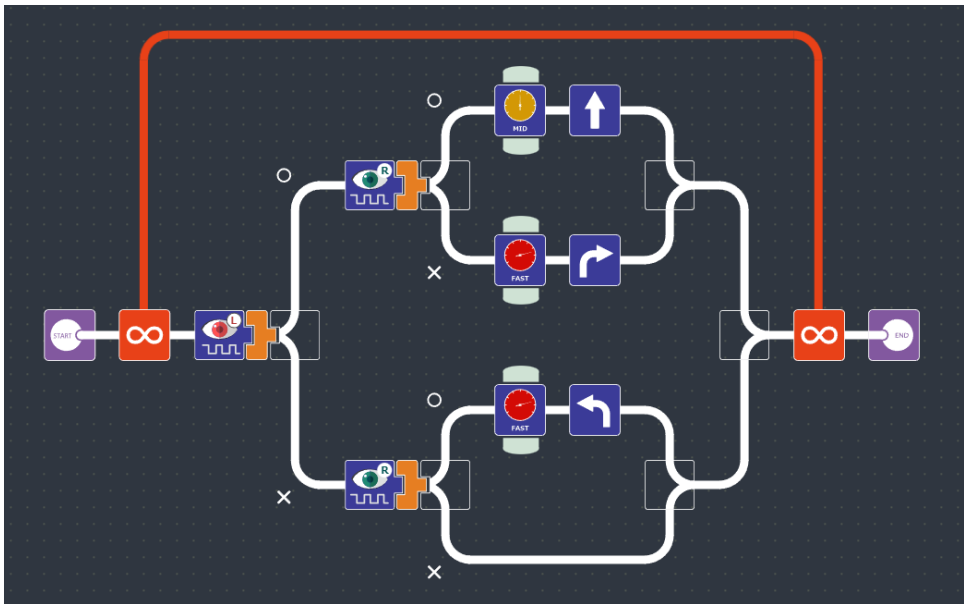


## やってみよう

センサーが外側になるように(中心から遠くなるように)取り付けて、ロボットが進むときの姿勢がどう変化するか観察してみましょう。

## 🔍 ライントレースする その2 左右のセンサーを使う

光センサー左と光センサー右の両方を使って、ラインに沿って進むプログラムを作成します。



(例6-3) ライントレースする その2

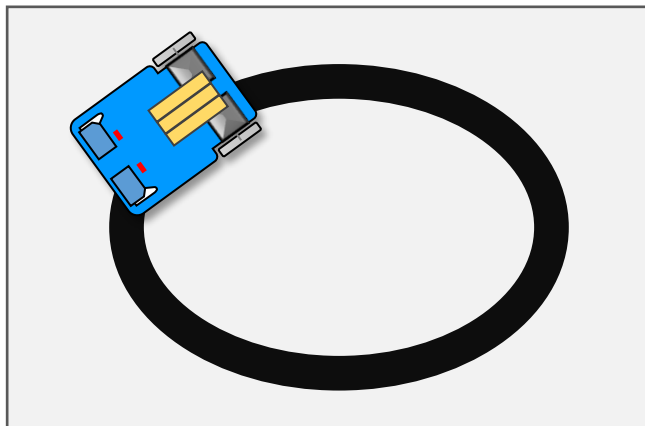
## 🔍 プログラムを観察してみよう

(例6-2)で光センサーのプログラムを作ったときは、光センサーの条件分岐で○と×の2通りだけ動きを設定しましたが、ここでは「光センサー(左)の条件分岐」の中にもうひとつ「光センサー(右)の条件分岐」が入っています。このように条件分岐の中にまた条件分岐を入れている構造のことを「**入れ子(ネスト)**」といいます。入れ子を使うことで条件分岐を3通り以上にすることができます。



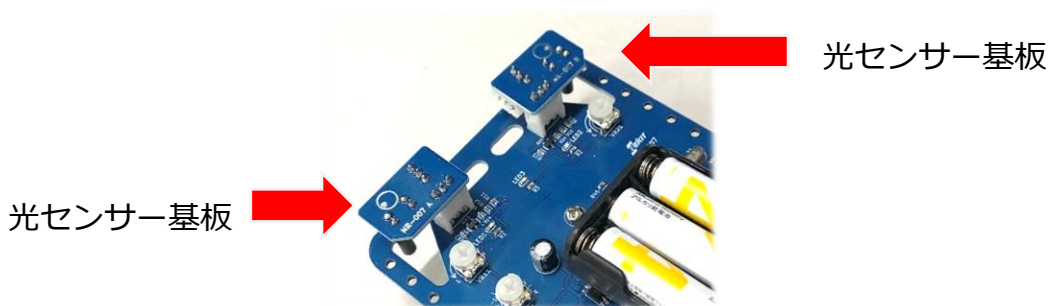
## 準備しよう

- ① ロボットを走らせるためのコースを準備しましょう。大きめの白い紙や床に、幅が広い黒テープなどで、下の図のようなコースをつくります。線の幅はできるだけ広い方が良く、2cm以上できれば3cm程度の幅を目安にします。



黒い線を作るときは、急カーブにならないようにつくります。  
急カーブがあるとそこでコースから外れやすくなります。

- ② ロボットの右側・左側のコネクタに光センサー基板が下を向くように、センサーどうしが遠くなるように差し込みましょう。



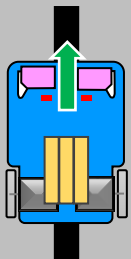
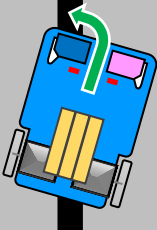
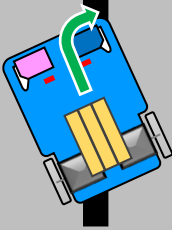
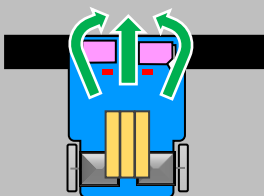
## 動作チェックをしよう

- 黒いラインの上をロボットが進み続けていますか？



## 動きを観察してみよう

ロボットの動きとセンサーの状態を表で整理してみましょう。

ライン (白・黒) の状態	光センサ左 (L)の状態	光センサ右 (R)の状態	ロボットの動き
	○ (白)	○ (白)	左右のセンサーの内側にラインがあるので <b>前進</b>
	× (黒)	○ (白)	左センサーの下にラインがあるので <b>左前進</b>
	○ (白)	× (黒)	右センサーの下にラインがあるので <b>右前進</b>
	× (白)	× (黒)	なにもしない (直前の動作を続ける)

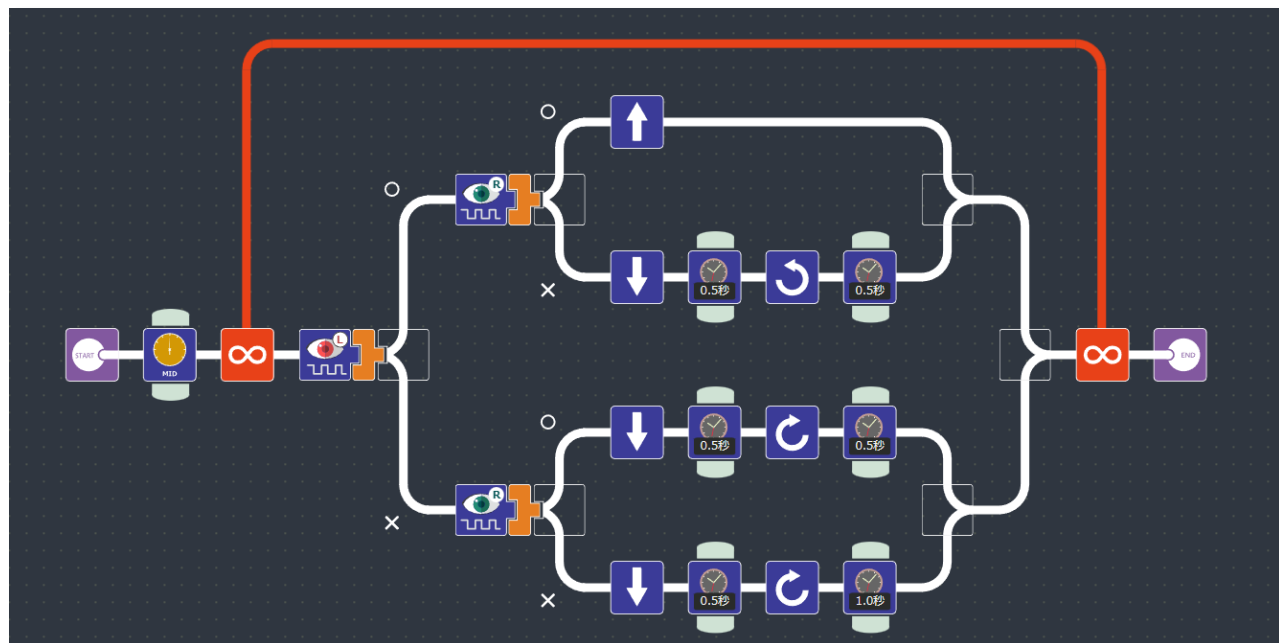
(例6-2)も同じ「ラインをトレースする」という目的を解決するプログラムなのに、センサーを2つ使うと動きがスムーズになったのが分かると思います。プログラムをつくる時は自分の使っている環境や状況に合わせて、いろいろな考え方で挑戦しましょう。





## 6-3 テーブルから落ちないロボットをつくる

光センサーを使って、テーブルの端(はし)を判別し、テーブルから落ちないロボットのプログラムを作成します。



(例6-3) 光センサーでテーブルの端(はし)を判別する



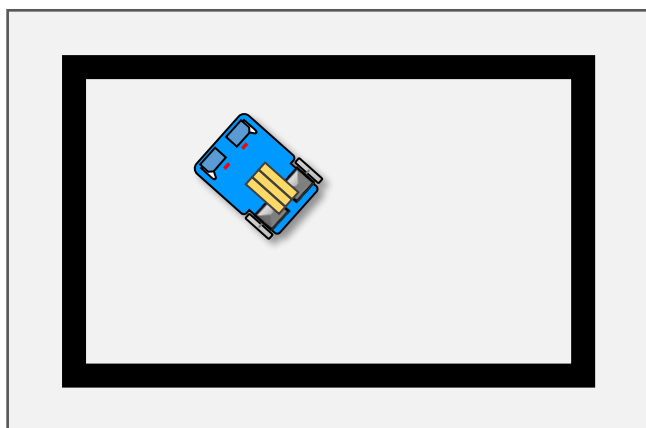
### プログラムを観察してみよう

(例6-3)と同じように、光センサーを2つ使っています。左右の光センサーのON/OFFの組み合わせでロボットの動作を変えたいので、条件分岐の中にさらに条件分岐をつくる「入れ子(ネスト)」を使ったプログラムになっています。

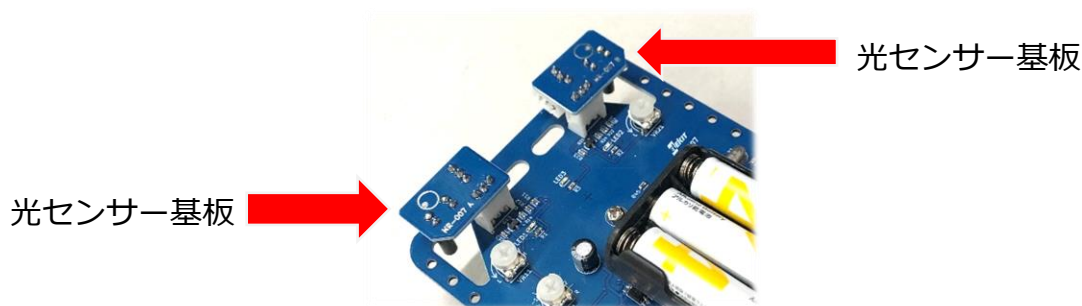


## 準備しよう

- ① ロボットを走らせるためのテーブルは、できるだけ白っぽいものを用意します。テーブルがない場合は、大きめの白い紙と、幅が広い黒いテープなどで、下の図のようなフィールドをつくります。



- ② ロボットの左側右側の両方のコネクタに光センサー基板が下を向くように、また、センサーどうしが離れた状態になるように差し込まれていることを確認しておきます。



## 動作チェックをしよう

- テーブルから落ちずに、または、黒いラインからはみ出さないようにロボットが動き続けていますか？



テーブルの上で動かす場合は、テーブルから落ちないように十分注意して動かしましょう。



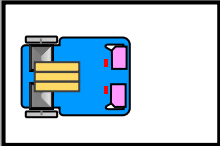
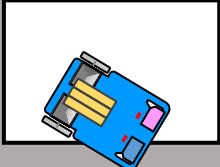

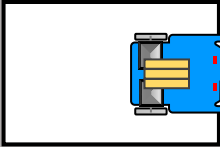
## 動きを観察してみよう

それでは、ロボットに取り付けた光センサーでどのようにしてテーブルの端を検知しているか、観察してみましょう。

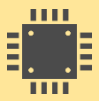
プログラム上で光センサーの条件分岐の○と×は、

○：光センサーが白を検知しているとき、つまりテーブルの上にいるとき

×：光センサーが黒を検知しているとき、つまりテーブルの端をみつけたときを意味します。センサーの状態とそのときのロボットの動きを表で整理してみましょう。

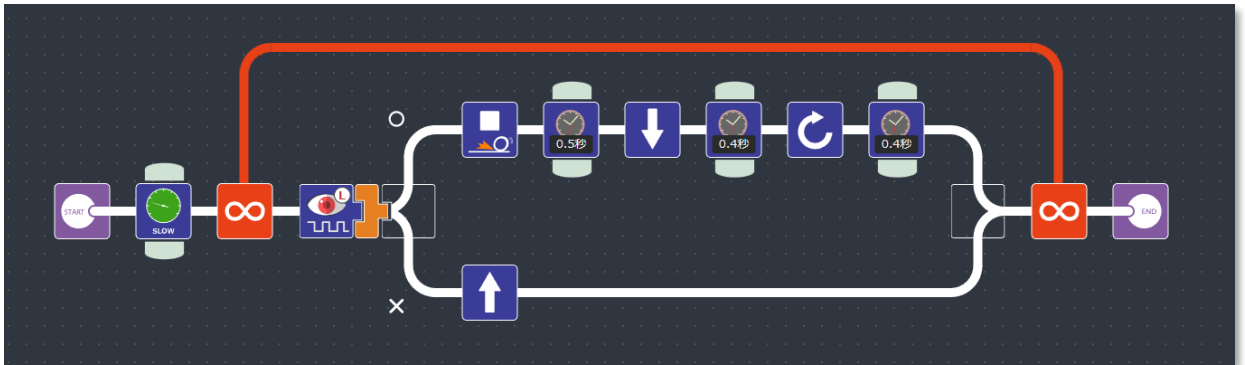
テーブルの状態	光センサ左(L)の状態	光センサ右(R)の状態	ロボットの動き
	○ (白)	○ (白)	両方のセンサーが テーブルの上なので <b>前進</b>
	○ (黒)	× (白)	右側のセンサーが テーブルの端なので <b>少し後進して 左回転</b>
	× (白)	○ (黒)	左側のセンサーが テーブルの端なので <b>少し後進して 右回転</b>
	× (黒)	× (黒)	両方のセンサーが テーブルの端なので <b>少し後進して 長めに右回転</b>

このようにして適切な動きをプログラミングすることで、ロボットはテーブルの端を判別して落ちないように動くことができます。



## 6-4 障害物をよけるロボットをつくる

ロボットの前方にある障害物を感知して、よけながら走るプログラムを作成します。

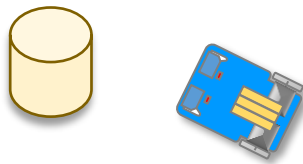


(例6-4) 障害物を感知してよける

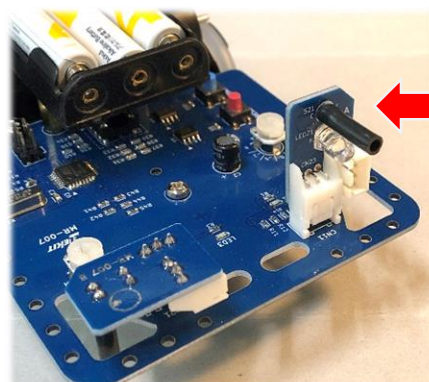


### 準備しよう

- ① ロボットを走らせる床に、障害物を置きましょう。障害物が黒い色だとうまく感知できないので、白っぽい色のものにします。



- ② ロボットの左側のコネクタに光センサー基板が前を向くように差し込みます。



光センサー基板



## 感度を調整しよう

これまで練習した6-1から6-3のロボットは、センサーを下向きに使っていましたが、ここではセンサーを正面に向けます。今までとセンサーの向きが変わりセンサーに入ってくる明るさも変わりますので、環境に合わせて感度を調整する必要があります。

### 上手な感度調整の方法

1. 感度調整ボリュームを反時計回りに回します。(感度が低い状態にします。) ボリュームは優しく回します。
2. ロボットを障害物などが何もない方向へ向けておいて、電源を入れます。センサー基板に搭載されている赤色のLEDが光ります。
3. この状態で本体基板上の出力表示LEDが点灯している場合は、周りが明るすぎるので、部屋をもう少し暗くします。
4. ロボットのセンサーの正面2～3cmのところに障害物を置きます。
5. ゆっくり感度調整ボリュームを時計回りに回していき、センサーが反応して出力表示LEDが点灯するところで回すのをやめます。
6. 障害物をロボットの正面から取り除いて、出力表示LEDが消えればOKです。
7. ロボットを別の方向に向けたときでも、障害物がないときは出力表示LEDが消えていて、障害物があるときは出力表示LEDが点灯することを確認しておきましょう。



## 動作チェックをしよう

感度調整ができれば、(例6-4)のプログラムを書き込みましょう。

- 障害物を見つけたら方向を変えてロボットが進み続けていますか？



## 動きを観察してみよう

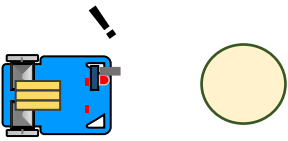

それでは、ロボットに取り付けた光センサーでどのようにして障害物を検知しているか、観察してみましょう。

プログラム上で光センサーの条件分岐の○と×は、

○：光センサーに入る光が多いとき、つまり正面に障害物があるとき

×：光センサーに入る光が少ないとき、つまり障害物がないとき

を意味します。センサーの状態とそのときのロボットの動きを表で整理してみましょう。

ロボット前方の状態	光センサ左(L)の状態	ロボットの動き
	○ (光の反射が多い)	前方に障害物がある <b>少し後進して 右回転する</b>
	× (光の反射が少ない)	障害物なし <b>前進</b>

このようにして適切な動きをプログラミングすることで、ロボットは障害物を見つけ、向きを変えて動くことができます。



## やってみよう

- もっとうまく障害物をよけることができるように、光センサーを2つ使ったプログラムにチャレンジしてみよう。

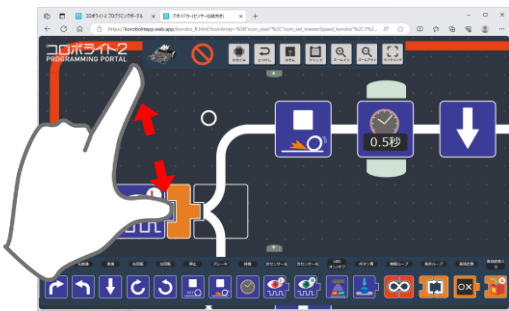
## 7. 覚えると便利な操作テクニック



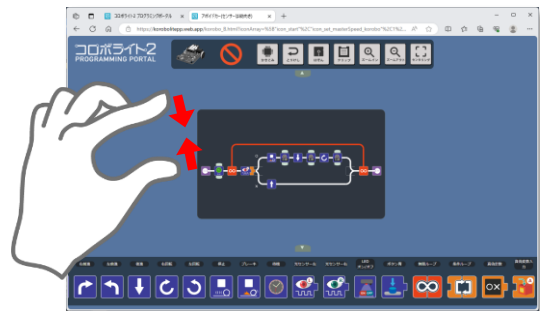
### 拡大・縮小

ピンチアウト・ピンチイン操作でプログラムの拡大・縮小をすることができます。

ピンチアウトで拡大



ピンチインで縮小



### 画面のスクロール・センタリング

プログラムを作成するとき、何も無い(アイコンが置かれていない)位置でドラッグ操作を行うとワークスペース全体のスクロール操作ができます。また、何も無い位置でダブルタップ(画面を軽くたたく操作を2回繰り返す)をするとワークスペースをセンタリングします。





### プログラムのクリップ

現在プログラムを表示しているブラウザー画面のURLアドレスを、クリップボードに保存します。


このURLをメールやメモ帳に貼り付けて他の人に伝えることで、別のパソコンのブラウザーで同じ画面が開きます。

## 8. アイコンの説明

### スタートアイコン、エンドアイコン








アイコン	アイコンの名前・説明
	<b>&lt;スタート&gt;</b> このアイコンからプログラムが始まります。 ワークスペースにあらかじめ配置してあり消すことはできません。
	<b>&lt;エンド&gt;</b> 全ての動作を停止し、プログラムを終了します。 ワークスペースにあらかじめ配置してあり消すことはできません。

### ドックのアイコン

アイコン	アイコンの名前・説明
	<b>&lt;前進&gt;</b> ロボットが前の方向へ進みます。
	<b>&lt;後進&gt;</b> ロボットが後ろの方向へ進みます。
	<b>&lt;左前進&gt;</b> ロボットが左斜め前に曲がりながら進みます。
	<b>&lt;右前進&gt;</b> ロボットが右斜め前に曲がりながら進みます。
	<b>&lt;左回転&gt;</b> ロボットがその場で左回転します。
	<b>&lt;右回転&gt;</b> ロボットがその場で右回転します。










## ドックのアイコン つづき

アイコン	アイコンの名前・説明
	<p><b>&lt;停止&gt;</b> ロボットのモーターの動作をOFFにします。</p>
	<p><b>&lt;ブレーキ&gt;</b> ロボットのモーターをショートさせて強制的に止まるようにブレーキをかけます。ピタッと停止させたいときなどに使います。</p>
	<p><b>&lt;待機&gt;</b> パラメータによって設定した時間プログラムの処理を待機させます。待機している間、直前の動作が継続されます。設定できる時間は最小0秒で、0.1秒ずつ増やすことができ最大で10秒です。初期値は1秒です。</p>
 	<p><b>&lt;光センサー (左 および 右)&gt;</b> 光センサーの状態を読み取り、真偽値で次のアイコンに渡します。光センサーがON状態(光センサーに光が十分に当たっている)の場合は [○] が、OFF状態(光センサーに光が十分に当たっていない)の場合は [×] が次のアイコンに渡されます。 アイコンのイラストに「L」と書かれているものが「左側コネクタに取り付けた光センサー」、「R」と書かれているものが「右側コネクタに取り付けた光センサー」です。</p> <div data-bbox="197 1170 1249 1377" style="border: 1px solid gray; border-radius: 15px; padding: 10px; margin: 10px 0;"> <p><b>真偽値とは？</b> 簡単にいうと、真は“正しい”、偽は“正しくない” という意味です。 プログラミング画面上では真は○、偽は× と表示されています。</p> </div>
	<p><b>&lt;ボタン青&gt;</b> STARTボタンの状態を読み取り、真偽値で次のアイコンに渡します。</p>
	<p><b>&lt;条件分岐&gt;</b> 真偽値を受け取り、[○]を受け取った場合は上の経路へ、[×]を受け取った場合は下の経路へ処理を進めます。 ※「条件分岐」アイコンはデータ(真偽値)を受け取るアイコンであるため、その左側には必ずデータ(真偽値)を渡すアイコンが配置されなくてはなりません。</p>

## ドックのアイコン つづき

アイコン	アイコンの名前・説明
	<p><b>&lt;スピード&gt;</b>            ロボットのモーターの回転するスピードをFAST(速く)かSLOW(遅く)その間のMIDの、どのスピードで回転させるか設定します。このアイコンを置いた後は次に設定するまで設定したスピードになります。            ※スピードアイコンを全く置かない場合のスピードはFASTです。</p>
	<p><b>&lt;LEDオン/オフ&gt;</b>            光センサー投光用LEDを点灯・消灯させます。</p>
	<p><b>&lt;無限ループ&gt;</b>            [無限ループ]内の処理を実行します。ループの最後に到達した場合は、[無限ループ]の先頭に戻ります。</p>
	<p><b>&lt;条件ループ&gt;</b>            真偽値を受け取り、[○]の場合は、[条件ループ]内の処理を実行しループの最後に到達すると、[条件ループ]の先頭に戻りもう一度真偽値を受け取ります。[X]の場合は、[条件ループ]末尾の次のアイコンに移動します。</p>
	<p><b>&lt;真偽定数&gt;</b>            パラメータボタンで設定した真偽値[○]または[X]を右のアイコンに渡します。</p>
	<p><b>&lt;真偽変数 入力&gt;</b>            真偽値を受け取り、パラメータボタンで指定した真偽変数(A, B, C, D, E)に格納します。</p>
	<p><b>&lt;真偽変数 出力&gt;</b>            パラメータボタンで指定した真偽変数(A, B, C, D, E)から真偽値を取り出して、右のアイコンに渡します。なお各真偽変数の初期値は[X]です。</p>
	<p><b>&lt;真偽反転&gt;</b>            真偽値を受け取り、その値を反転させた真偽値を右のアイコンに渡します。[○]を受け取ると[X]を渡し、[X]を受け取ると[○]を渡します。</p>
	<p><b>&lt;論理演算&gt;</b>            [論理演算]アイコンの左側と内側の真偽値で論理演算を行った結果を、真偽値で右のアイコンに渡します。論理演算の種類(AND, OR, XOR)は、パラメータボタンで選択します。</p>

## ドックのアイコン つづき

アイコン	アイコンの名前・説明
	<p><b>&lt;回数ループ&gt;</b>            数値を受け取り、その回数、[回数ループ]内の処理を実行します。</p>
	<p><b>&lt;数値定数&gt;</b>            パラメータボタンで設定した数値を右のアイコンに渡します。パラメータボタンをクリックまたはドラッグして数値を設定します。設定できる数値は最小-100、最大100の範囲内の整数です。</p>
	<p><b>&lt;数値変数 入力&gt;</b>            数値を受け取り、パラメータボタンで指定した数値変数(A, B, C, D, E)に格納します。格納できる数値の範囲は、-32768 ~ 32767の整数となります。</p>
	<p><b>&lt;数値変数 出力&gt;</b>            パラメータボタンで指定した数値変数(A, B, C, D, E)から数値を取り出して、右のアイコンに渡します。なお各数値変数の初期値は0です。</p>
	<p><b>&lt;数値演算&gt;</b>            [数値演算]アイコンの左側と内側で数値をそれぞれ受け取り、左側と内側の数値で数値演算を行った結果を、数値で右のアイコンに渡します。数値演算の種類(足し算、引き算、掛け算、割り算)は、パラメータボタンで選択します。なお割り算で内部のアイコンの数値が「0」のときは、実行時に「1」に置き換えられます。</p>
	<p><b>&lt;正負反転&gt;</b>            数値を受け取り、その値の正負(プラスマイナス)を反転させた数値を右のアイコンに渡します。</p>
	<p><b>&lt;数値比較&gt;</b>            [数値比較]アイコンの左側と内側の数値で数値比較を行った結果を、真偽値で右のアイコンに渡します。数値比較の種類(大なり(&gt;), 小なり(&lt;), 等価(==))は、パラメータボタンで選択します。</p>



# 論理演算、数値演算、数値比較

ここでは、論理演算、数値演算、数値比較アイコンについて、補足説明します。



論理演算



数値演算



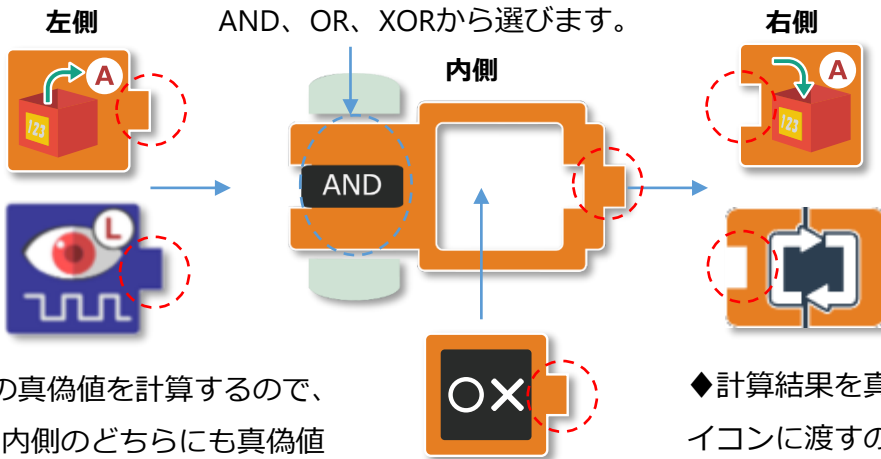
数値比較



## 論理演算アイコン

「論理演算」アイコンは、左側と内側においた2つの真偽値を使って計算した結果を、やはり真偽値で次のアイコンに渡します。計算の種類にはAND(アンド)、OR(オア)、XOR(エックスオア)があります。AND、OR、XORそれぞれの左側と内側の真理値の組み合わせと計算の結果は次ページの真理値表のとおりになります。

◆計算の種類は上下のボタンで

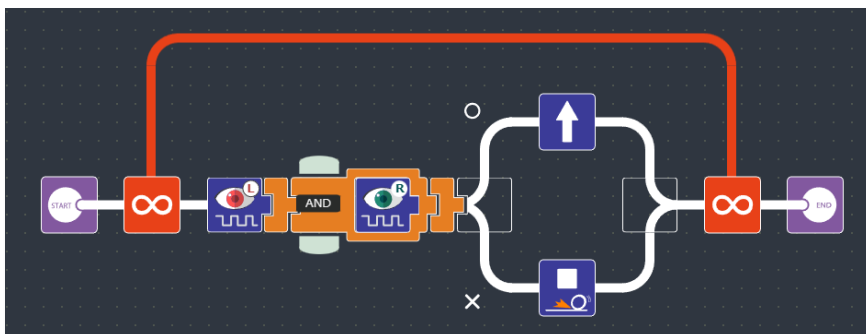


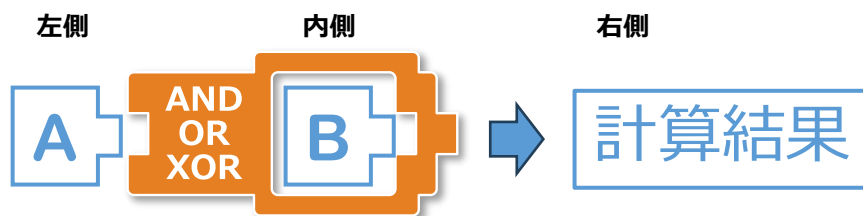
◆2つの真偽値を計算するので、左側と内側のどちらにも真偽値を渡すアイコンをおきます。

◆計算結果を真偽値として次のアイコンに渡すので、右側に真偽値を受け取るアイコンをおきます。

次のプログラムは、「光センサー左」アイコンと「光センサー右」アイコンの状態をANDで計算して、その結果が真のときだけモーターを動かします。

光センサー左/右の状態の組み合わせで、モーターがどう動くか試してみましょう。





## **AND (論理積 : 両方とも真のときだけ、結果は真)**

論理演算アイコンの左側と内側に配置したアイコンが両方とも真ならば、右側につないだ次のアイコンに真を渡し、それ以外の場合は偽を渡します。

A	B	計算結果
偽 (×)	偽 (×)	偽 (×)
偽 (×)	真 (○)	偽 (×)
真 (○)	偽 (×)	偽 (×)
真 (○)	真 (○)	真 (○)

上の表は**真理値表**といい、入力の組み合わせと対応する結果を表にしたものです。

## **OR (論理和 : どちらか1つでも真のとき、結果は真)**

論理演算アイコンの左側と内側に配置したアイコンのどちらか1つでも真ならば、右側につないだ次のアイコンに真を渡し、それ以外の場合は偽を渡します。

A	B	計算結果
偽 (×)	偽 (×)	偽 (×)
偽 (×)	真 (○)	真 (○)
真 (○)	偽 (×)	真 (○)
真 (○)	真 (○)	真 (○)

## **XOR (排他的論理和 : どちらか1つだけが真のとき、結果は真)**

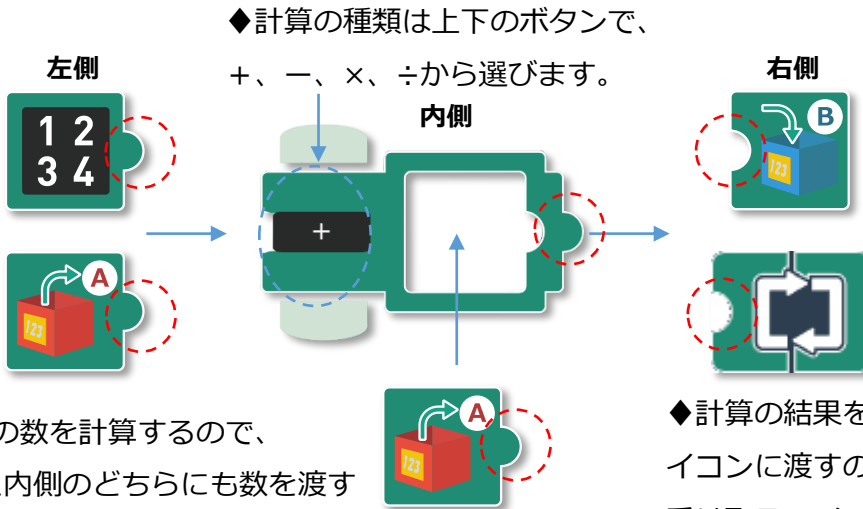
論理演算アイコンの左側と内側に配置したアイコンのどちらか1だけが真ならば、右側につないだ次のアイコンに真を渡し、それ以外の場合は偽を渡します。

A	B	計算結果
偽 (×)	偽 (×)	偽 (×)
偽 (×)	真 (○)	真 (○)
真 (○)	偽 (×)	真 (○)
真 (○)	真 (○)	偽 (×)



## 数値演算

「数値演算」アイコンは、左側と内側においた2つの数を使って、「足し算」、「引き算」、「かけ算」、「わり算」の計算をします。計算した結果の数を次のアイコンに渡します。



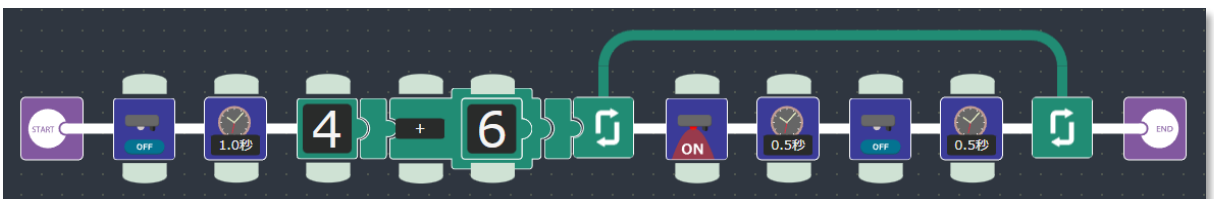
足し算、引き算、かけ算、わり算については、算数や数学の計算と同じですが、コロボライト2の処理能力の関係で、次のような制限があります。

演算の結果は、-32768 ~ 32767 の範囲の整数となる

このため、計算結果が小数になったときは、小数点以下は切り捨てられます。  
また、計算結果が -32768 より小さくなったときは「-32768」となります。  
計算結果が 32767 より大きくなったときは「32767」となります。

次のプログラムは、2つの「数値定数」アイコンの値を計算して、その結果の数値をLEDの点滅の回数で表示します。

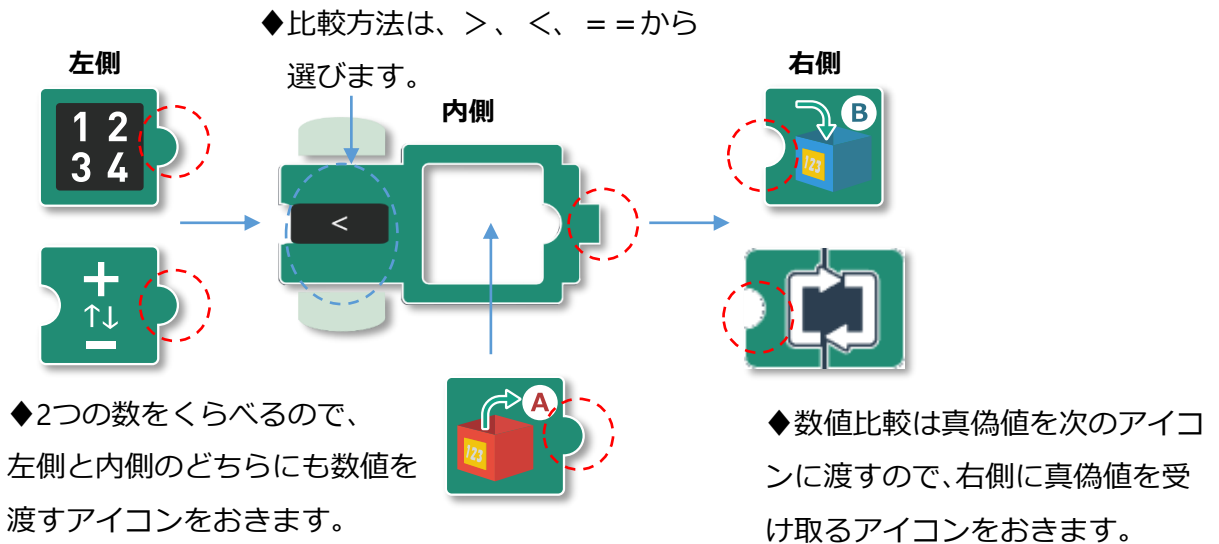
「数値定数」アイコンの値や、計算の種類を変えてみて、LEDの点滅回数がどうなるかを確かめてみましょう。





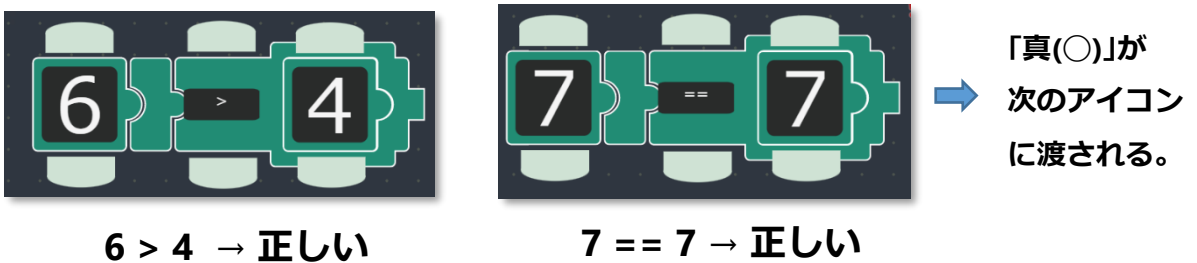
## 数値比較

「数値比較」アイコンは、左側と内側に置いた2つの数の大きさをくらべます。算数の計算のように、 $>$  (左側の数大きい)、 $<$  (左側の数小さい)、 $==$  (同じ)で2つの数をくらべる式を作ります。その式が正しいければ真、正しくなければ偽を次のアイコンに渡します。

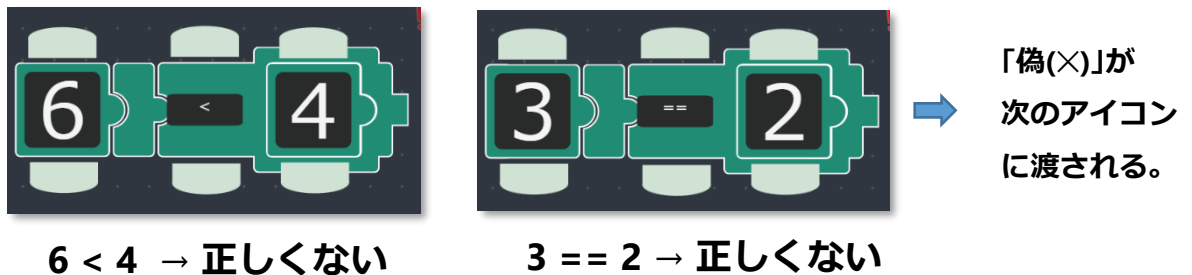


「数値比較」アイコンは、接続したアイコンと選択した計算の種類で作った式が「正しい(O)/正しくない(X)」を真偽値として次に渡します。

例えば、次の式は「正しい」ので、「真(O)」が次の右側につないだアイコンに渡されます。



次の式は「正しくない」ので、「偽(X)」が次の右側につないだアイコンに渡されます。



## 9. トラブルシューティング

作成したプログラムが書き込めない。



- 充電専用USBケーブルを誤って使っていないか確認します。プログラムの転送には付属のUSBケーブルを使います。
- ロボットの電源スイッチがONの位置になっていることを確認します。
- ロボットのリセットボタンを押してみます。
- USBケーブルを一旦抜いて、差し込みなおしてみます。
- パソコン側のUSB端子が不調である可能性があります。パソコンに別のUSB端子があれば、そちらの端子を使って確認してみます。
- ペアリングをもう一度行ってみます。すでに「ペア設定済み」と表示されていても**キャンセルせずに、もう一度[KOROBO LITE 2]を選択してOKを押します。**

保存したプログラムが見つからない。

- プログラムを保存したときに使用したブラウザーであることを確認します。
- ブラウザーにログインしているユーザーが、保存したときと同じユーザーであることを確認します。
- ※ ブラウザーが同一であっても、ログインユーザーが違っているとプログラムは表示されません。
- ※ ブラウザーのシークレットモードを使っていた場合はプログラムは保存されていません。