

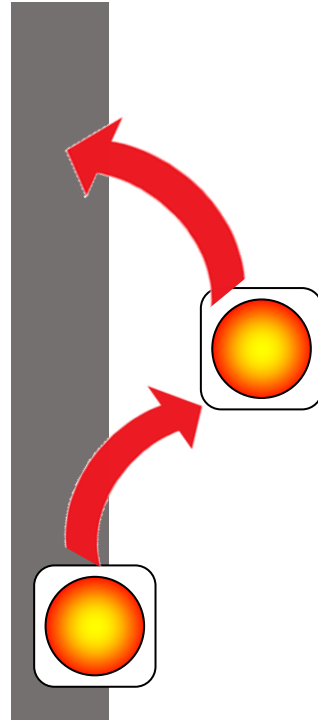
3) なめらかな動きをしよう! 名前()

1. ライントレースをしよう

① シグザグトレース (ONOFF 制御) 黒線の右側をトレース

```

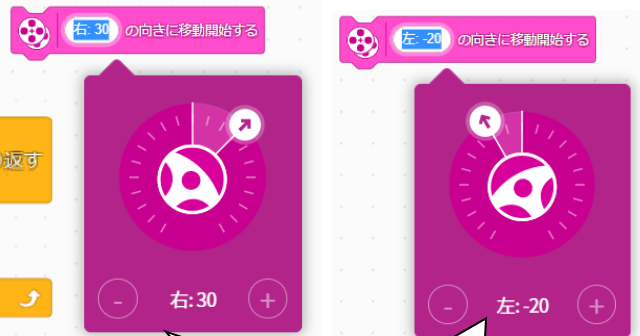
    プログラムがスタートしたとき
    移動に使うモーターを A+B にする
    モーター1回転を 17.5 cm にする
    移動スピードを 50 % にする
    B 相対位置を 0 に設定
    700 < B 相対位置 まで繰り返す
    もし E 反射光が < 50 % なら
    右: 30 の向きに移動開始する
    でなければ
    左: -30 の向きに移動開始する
    移動をやめる
    すべて の停止
  
```



② なめらかトレース (Power 制御) 目標との明るさの差をステアリングに入力

```

    定義 H-tra cm
    B 相対位置を 0 に設定
    cm / 5 * 100 < B 相対位置 まで繰り返す
    E の反射光 - 50 の向きに移動開始する
    移動をやめる
  
```



目標からずれた分だけハンドルを切る。 → 線にもどる。

どうして線をたどれるの?

ロボットが黒か白かを見分ける目印になる数値をしきい値といえます。

たとえば白が100、黒が20のとき、両方の値を足して2でわったものを入れておくのが普通です。

50-反射光→ステアリング

反射光の明るさが50の時	$50 - 50 = 0$ (直進)
20など暗い時	$50 - 20 = 30$ (右方向)
70など明るい時	$50 - 70 = -20$ (左方向)

それぞれマイブロックにしておくともよいでしょう。

【発展】
線の左側もトレースできるかな?

③ マイブロックにする。(計算でcm制御)



マイブロックに入力を追加

プログラムの動きを調整するための「入力」を仕込んでおくと便利です。

わかりやすい名前を付けておきましょう。

例では、SP…スピード
cm…長さ (cm)
としています。

いつまで繰り返すかの「判断ブロック」では、cmに20.6をかけて角度に変換しています。

タイヤの一周が 17.5cm。一周が 360度なので、
 $360 \div 17.5 = 20.57\dots$
なので 1cmあたり 20.6度として計算しています。

6年生ならわかるかな？

④ 前回の続きで赤い音符を得点エリアに入れよう！



実際の動き

⑤ サンプルプログラム

定義 Sime アームを閉めるプログラム

定義 Ake アームを開けるプログラム

定義 H-tra SP cm なめらかライトレース



となりのオブジェクトまで飛ばしてしまうので、アームを小さくしました。