

# 歩くロボット

前へ進むロボットを組み立てよう — でも車輪を使わないこと!



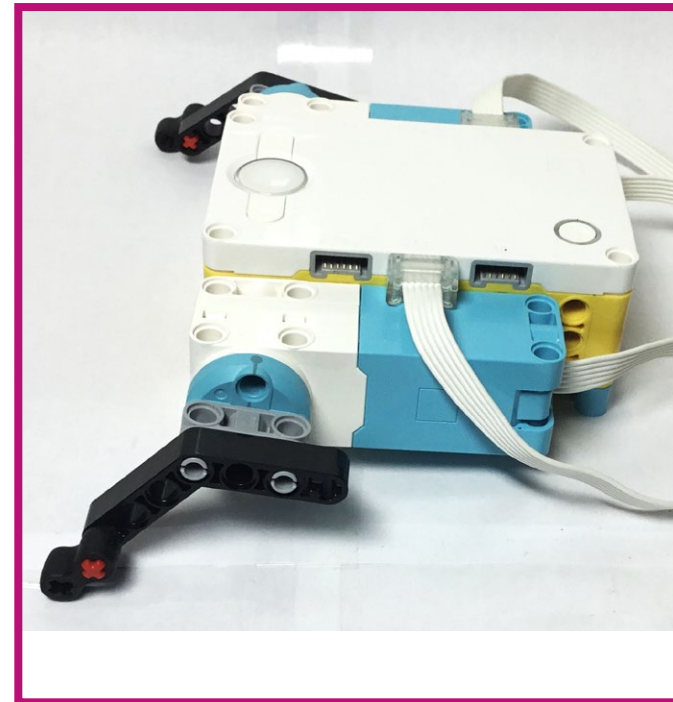
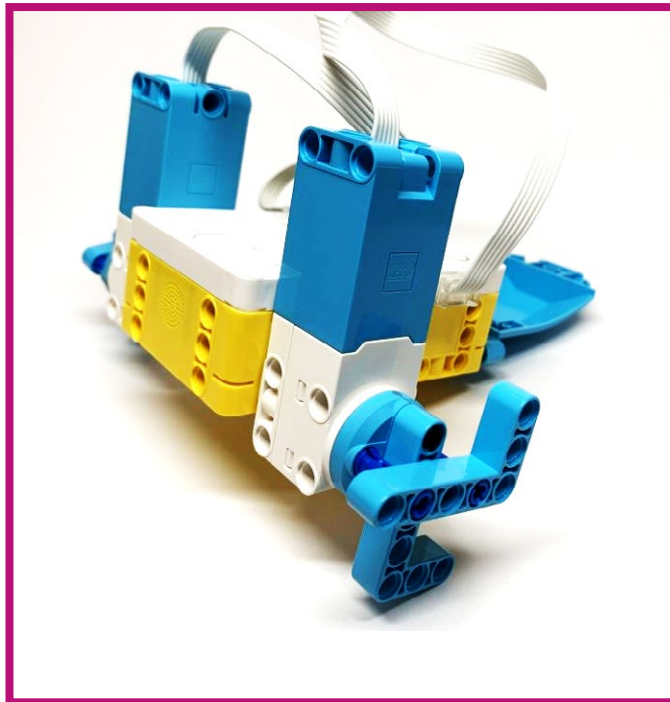
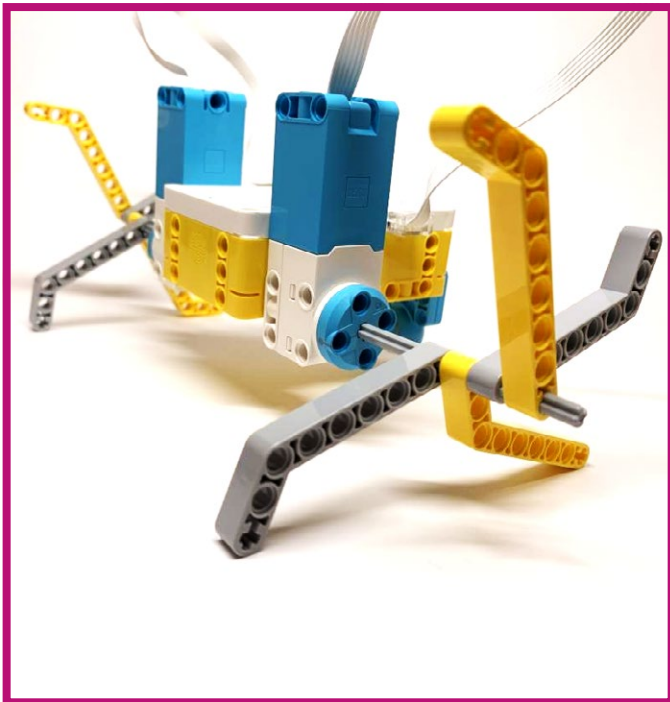
エンジニアのつもりで考えよう:  
車輪をつけずにロボットを前に進ませるには?

物理学者のつもりで考えよう:  
ロボットの足の長さによって動きが変わる?



## サンプル

モーターに「足」をつける方法をいくつか考えてみよう。



カードをうら返してくわしく見てみよう!





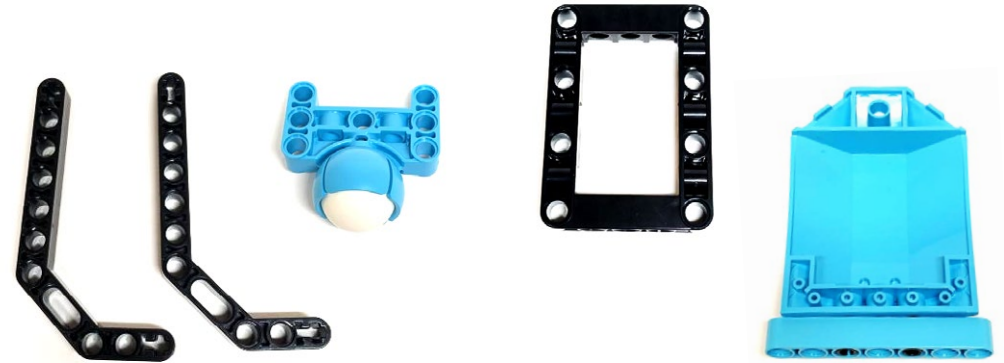
## 組み立てよう!

レゴパーツの使い方は工夫次第! 思いきった新しい発想でクリエイティブに考えて、使うパーツを選ぼう!

足として使えるパーツ:



安定そう置や支柱として使えるパーツ:



## プログラミングしよう!

1回転ごとにロボットが止まるようにしよう。  
どれくらいの速さでロボットを動かしたい?

```
1 from hub import port
2 import runloop, motor_pair, time
3
4 motor_pair.unpair(motor_pair.PAIR_1)
5 motor_pair.pair(motor_pair.PAIR_1, port.A, port.B)
6
7 async def main():
8     for i in range(10):
9         await motor_pair.move_for_degrees(motor_pair.PAIR_1, 100, 0, velocity=500)
10        time.sleep_ms(1000)
11
12 runloop.run(main())
```

## アレンジしてみよう:

- センサーを使って、ロボットがかべにぶつからないようにしよう。
- 2つのモーターをいっしょに動かす? それとも交代で動かす?



## チャレンジしよう!

モーターを1つだけ使って歩くロボットを作ってみよう。

# みなさん、こんにちは

みんなにあいさつしてくれるロボットを組み立てよう。手をふる、ゲータッチ、ハイタッチ…他にはどんなあいさつの動作があるだろう？



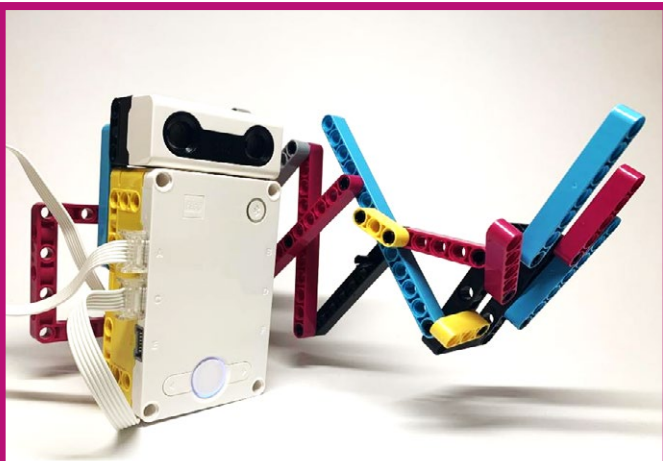
**エンジニアのつもりで考えよう:**  
モーターをどこにつければ、うでの動きがもっと自然になる？



**アーティストのつもりで考えよう:**  
どうすればもっと親しみやすいフレンドリーなロボットになる？

## サンプル

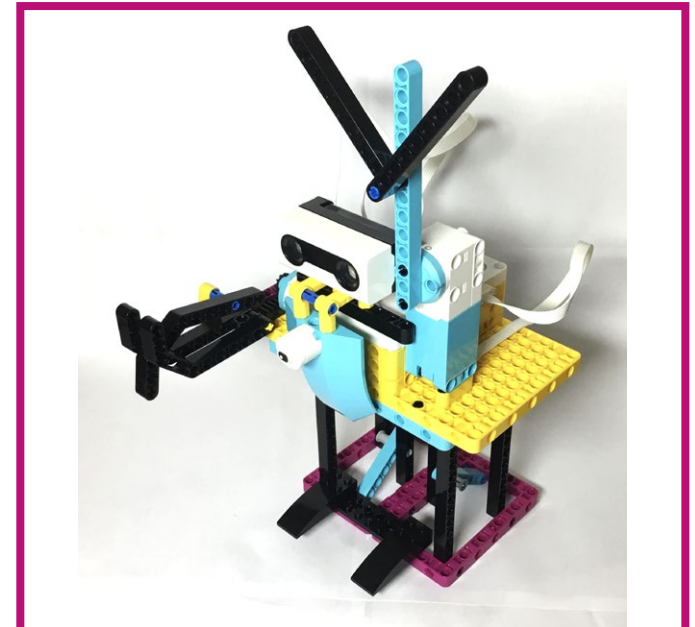
どんなあいさつをするロボットにしたい？



ハイタッチ ロボット



手をふるロボット



ゲータッチ ロボット

カードをうら返してくわしく見てみよう！



初級



総合スキル



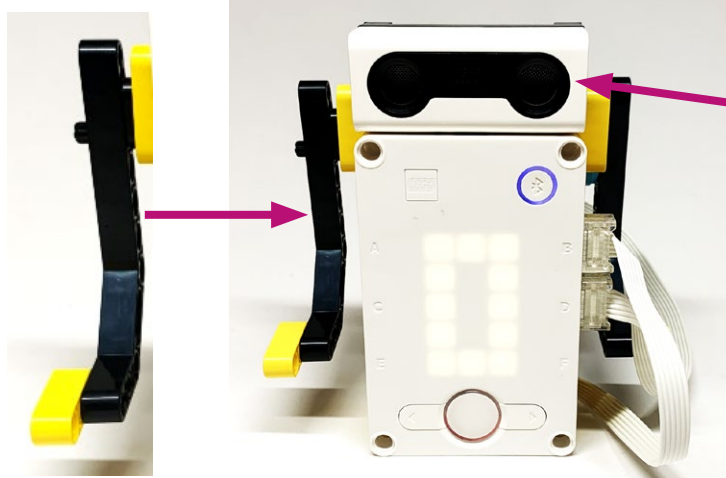
仕組みと動き





## 組み立てよう!

ロボットにどんなパーツをつけたい?



うでの先にパーツを取り付けて、手のようにしよう。



ロボットの目みたいに見える  
ちょう音波センサーをつけたら、動きを自動化できる。



「指」をつければ、手がうんとリアルになる。

## プログラミングしよう!

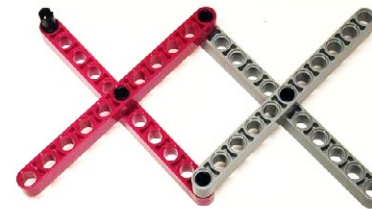


ロボットをどれくらい動かしたいか考えよう。動きの量は、時間で  
はかる?それとも回転数で?

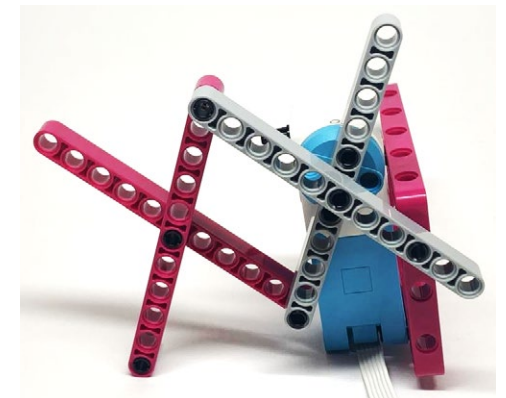
```

1 from hub import port
2 import runloop, motor
3
4 async def main():
5     for i in range(10):
6         await motor.run_for_degrees(port.A, 180, 500)
7         await motor.run_for_degrees(port.A, -180, 500)
8
9 runloop.run(main())

```



パーツをX型にしてつなげれば、のびちぢみするアームができる。



## チャレンジしよう!

ちょう音波センサーを使って、だれかが目の前に来たときだけロボットが動くようにしよう。

# 人形げき

SPIKE™ プライムを使って動く人形を自由に作ろう。



## エンジニアのつもりで考えよう:

はじめに作った人形をもっとよいものにするために、どんなテストをして、どこを改良する？

## 生物学者のつもりで考えよう:

どんな生き物を作る？その生き物はどんなふう動く？

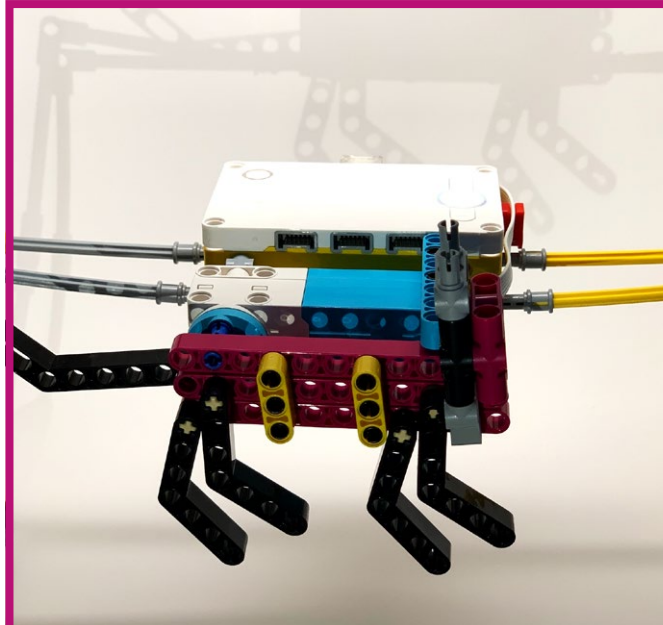


## サンプル

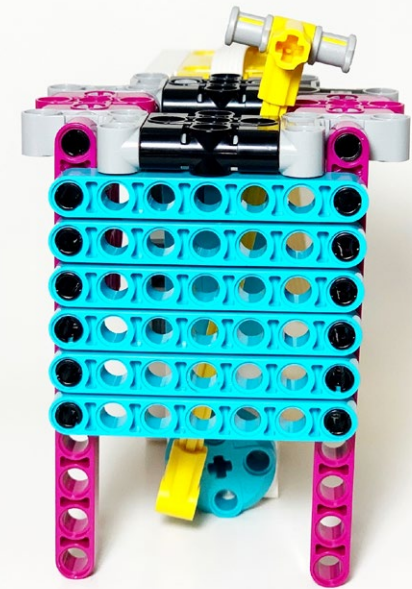
いろいろな種類の人形を作ってみたよ。他にも思いついたかな？



人型のあやつり人形



ネコ型のかげ絵人形



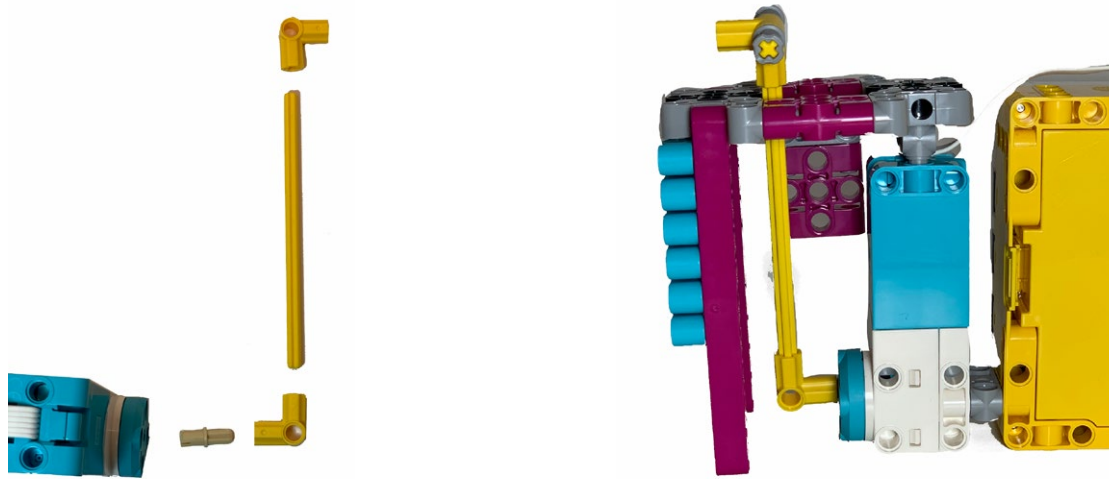
カゴに入ったヘビのぼうつかい人形

カードをうら返してくわしく見てみよう！



## 組み立てよう!

どうやったら回転運動を直線運動に変えられるか、考えてみよう。  
好きな動物で人形を作ってみよう!



## プログラミングしよう!

おどる あやつり人形のプログラム

```
1 from hub import port
2 import runloop, motor
3
4 async def main():
5     count = 0
6
7     while count < 10:
8         await motor.run_for_time(port.D, 500, -300)
9         await motor.run_for_time(port.D, 1000, 300)
10        await motor.run_for_time(port.D, 1000, 300)
11
12        count += 1
13
14 runloop.run(main())
```

## アレンジしよう

- 別のボタンをおしたら人形がちがう動作をするように変えられるかな?
- 動作と同時に音を鳴らすようにできる?
- 速度を変えると、人形の動作はどんなふうになる?



## チャレンジしよう!

音楽に合わせて動く人形を作ってみよう。



# お庭

お庭の草花はどんなふう to 育つ? キミのお庭には何がある? — 満開のお花? 大きく育つ元気なお野菜? ロボットじかけの熊手?



## 庭師のつもりで考えよう:

お庭には、どんな草花があるだろう? 生き物以外のアイテムはどうかな?

## エンジニアのつもりで考えよう:

複数のモーターを使って、お庭に動きをつける方法を考えよう。



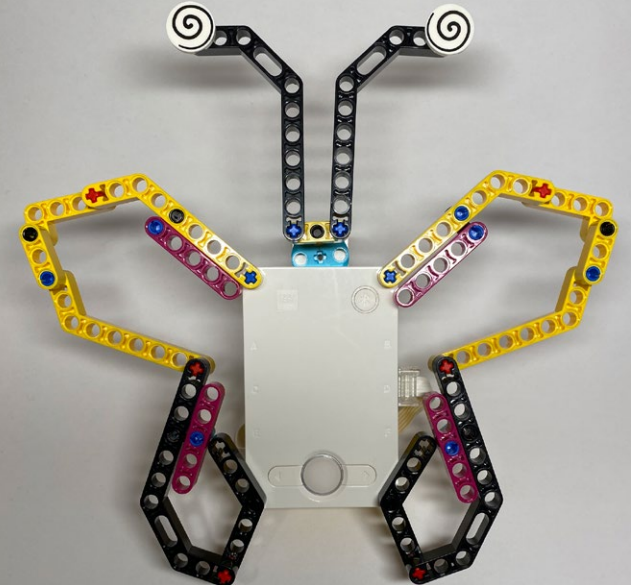
## サンプル



満開のお花



しあわせな巣箱



おとぎの国のチョウチョ

カードをうら返してくわしく見てみよう!



初級



総合スキル



自然と動物



## 組み立てよう!

パーツの形をよく見てみればアイデアが浮かぶかな? リアルな草花やお庭にあるアイテムを組み立てよう!



← ヒマワリの花びらは、黄色のまっすぐなパーツと曲がったパーツでできているよ →



## プログラミングしよう!

チョウチョのしよっ角を動かすプログラム:



```
1 from hub import port
2 import runloop, motor
3
4 async def main():
5     initial_power = 100
6     rotation = 0.5
7     for i in range(10):
8         await motor.run_for_degrees(port.A, int(rotation * 360), initial_power)
9         await motor.run_for_degrees(port.A, int(rotation * -360), initial_power)
10
11 runloop.run(main())
```

## アレンジしよう:

- 組み立てモデルはどれも世界にひとつだけだから、モデルに合わせてモーターの回転のあたいを変える必要がある。
- (プログラムのはじめだけ、あるいはいろいろなタイミングで) モーター速度を変えると、草花やアイテムをもっといきいきと動かせるようになる。
- ブロックが光るパターンを変えてみよう!



## チャレンジしよう!

センサーを使って、お庭のものが動き始めるようにできるかな?



# シンプルなSPIKE™車

できるだけ少ないパーツで がんばりな車を組み立てよう。ロボットカーのプログラミングは「2秒間前へ進む・2秒間後ろに下がる・2秒間右に回る・2秒間左に回る」という動きができるようにするのが条件だ。



## エンジニアのつもりで考えよう:

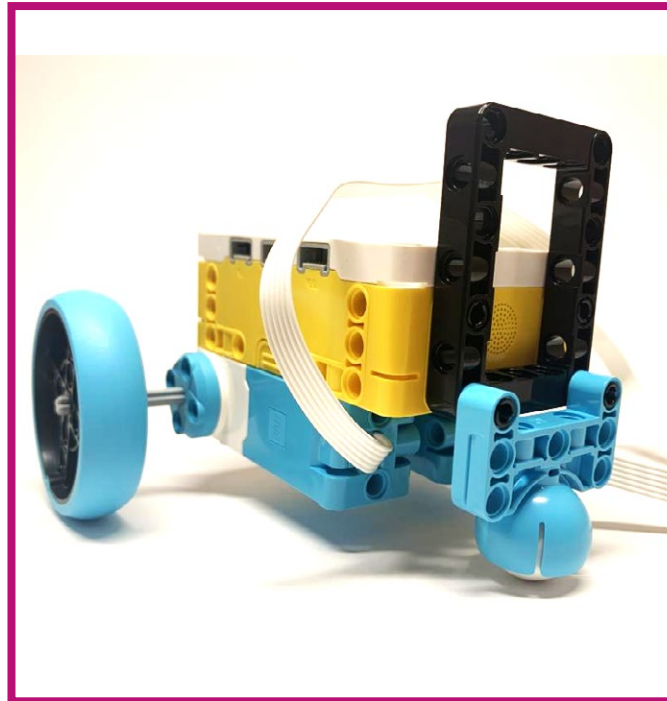
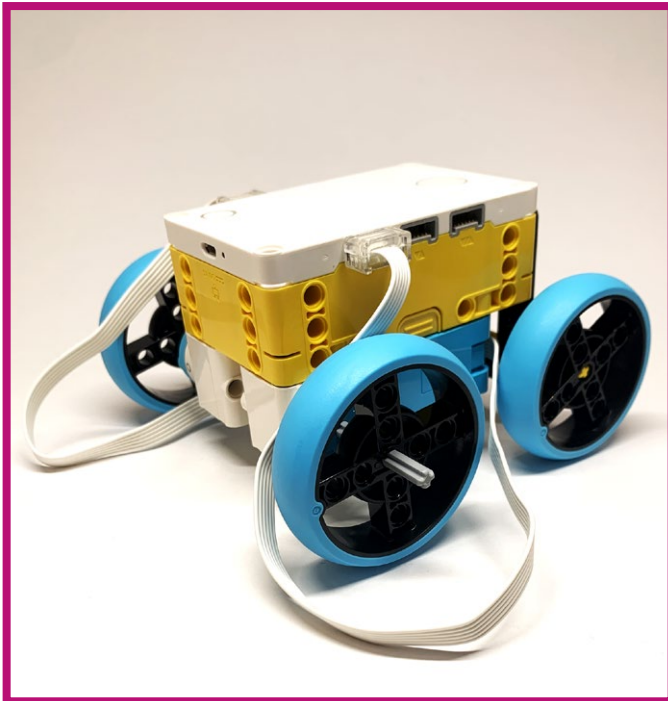
ロボットカーがどうしたらうまく動くか、いろいろ自由に試してみよう!

## 建築家のつもりで考えよう:

いくつかのパーツをまとめて別のパーツと置きかえて、数を減らせる?



## サンプル



## バランスをとろう

ロボットカーの重心をうまく調整して、車輪だけで自立するようにしよう。

これはダメ!

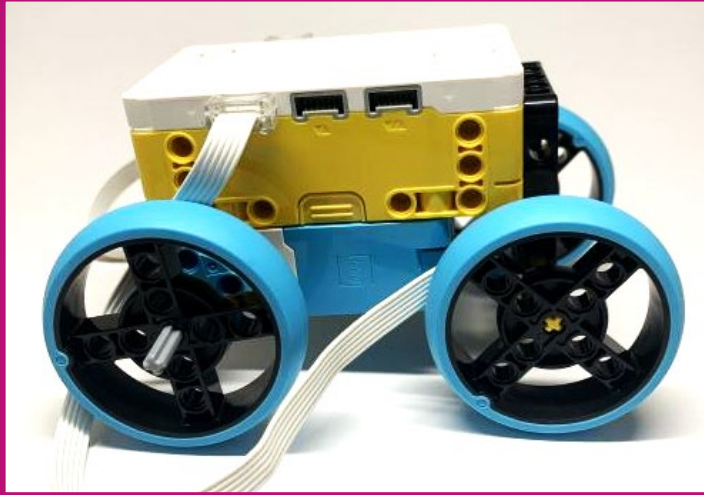
カードをうら返してくわしく見てみよう!



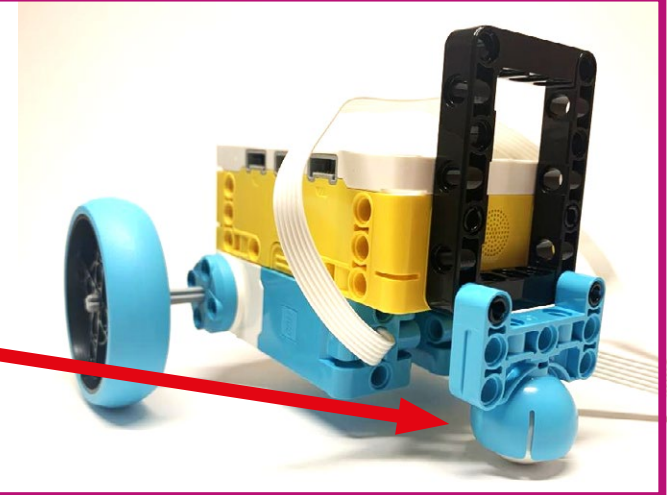


## 組み立てよう!

いろんな仕組みの車を作れるかな?



このキャスター式車輪は、どんな方向へも曲がれるから便利!



## プログラミングしよう!

2つのモーターが前へ進むプログラム:



```
1 from hub import port
2 import runloop, motor_pair
3
4 motor_pair.unpair(motor_pair.PAIR_1)
5 motor_pair.pair(motor_pair.PAIR_1, port.A, port.B)
6
7 async def main():
8     await motor_pair.move_tank_for_time(motor_pair.PAIR_1, 500, 500, 2000)
9
10 runloop.run(main())
```

## アレンジしてみよう!

このプログラムをもとに考えよう: どう変えれば、ロボットカーが後ろに下がって方向をかえるようにできるかな?



## チャレンジしよう!

ロボットカーを8の字走行させてみよう!

# 打楽器で遊ぼう

ジャンジャン・ドンドン・バーン! バンドのために打楽器を作ろう。たいこ・シンバル・木きん・ベル…他には何を思いつく? どんな打楽器を作ってもいいよ。



エンジニアのつもりで考えよう:  
もっと大きな音を出すために、デザインをどう改良する?

ミュージシャンのつもりで考えよう:  
思いつくかぎりの打楽器をあげてみよう。



## サンプル

モーター以外のものを取り付けて音を出せるか、試してみよう!



マラカス



たいこ



木きん

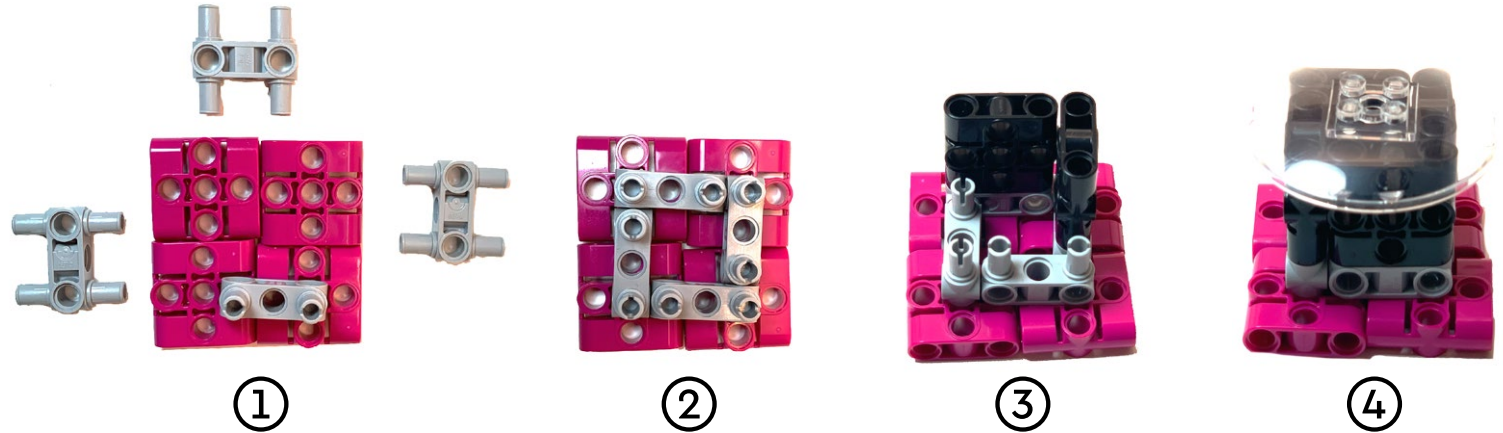
カードをうら返してくわしく見てみよう!



## 組み立てよう!

かんたんな組み立て (箱など) から始めて、パーツを付け足して複雑なもの (箱の中にレゴパーツを入れてマラカスのように音が出るようにするなど) にしていこう。

かんたんな組み立てといっても、組み立て方はたくさんあるんだ!



## プログラミングしよう!

マラカスをふるプログラム:



```

1 from hub import port, sound
2 import runloop, color_sensor, color
3
4 async def main():
5     while True:
6         colorSeen = color_sensor.color(port.E)
7
8         if colorSeen == color.BLACK:
9             await sound.beep(800, 500, 50)
10
11        if colorSeen == color.AZURE:
12            await sound.beep(1200, 500, 50)
13
14        if colorSeen == color.YELLOW:
15            await sound.beep(1400, 500, 50)
16
17 runloop.run(main())

```

## 考えるポイント!

- お気に入りの曲に合うように、マラカスをふるタイミング (時間のあけ方) を変える。
- モーターの速度を変えると、何が変わる?
- マラカスをふる時間の長さを長く / 短くするためには、プログラムをどんなふうに変えればいい?



## チャレンジしよう!

いっしょに歌うなら、リズムだけじゃなくメロディーもほしい。決まった曲をえんそうできる楽器を作れるかな?



# うちゅうたんさく

うちゅう旅行へ出発!うちゅうたんけんにまつわるアイテムといえば何を作る?ロケットやうちゅう飛行士が使う道具から、エイリアンまでいろいろある。



## 発明家のつもりで考えよう:

もしうちゅうに行くことになったら、どんな道具を持っていったら便利だと思う?

## たんけん家のつもりで考えよう:

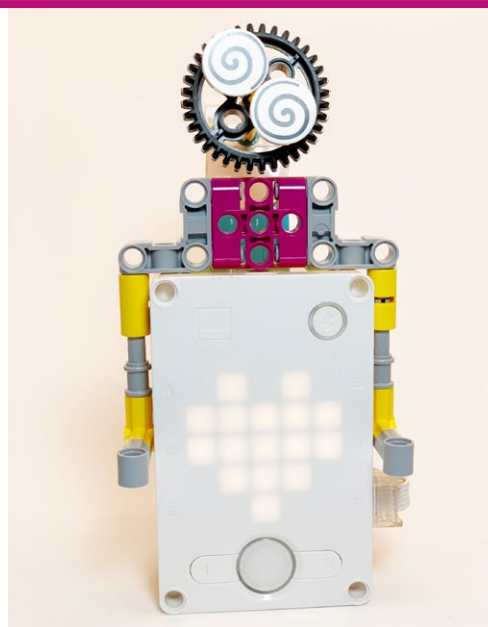
うちゅうのさまざまな星で出会う生き物や自然を想像しよう。



## サンプル



発しゃするロケット



頭が回るエイリアン



公転するわく星

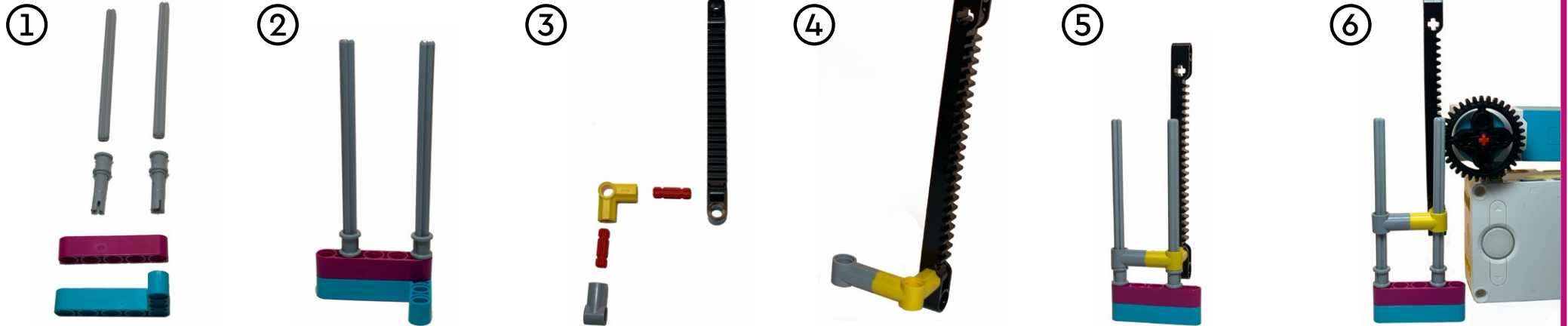
カードをうら返してくわしく見てみよう!





## 組み立てよう!

ラック (直線形の歯車) とピニオン (円形の小さな歯車) を組み合わせると、(ロケットのサンプルのように) モーターの回転運動を直線運動に変えられる。ピニオンが車輪のようにラックにそって転がることで、ラックが前後/上下に動く。



ラックが上下にスムーズに動く土台の例

## プログラミングしよう!

エイリアンのプログラム:



```

1 from hub import button, port, light_matrix
2 import runloop, motor
3
4 async def main():
5     while True:
6         if button.pressed(button.LEFT):
7             light_matrix.write("Blast Off!", 100, 250)
8             await motor.run_for_degrees(port.A, -90, 100)
9
10 runloop.run(main())

```

## アレンジしよう

- ハブを使って、うちゅうの音を鳴らせる?
- ハブのライトをつけられる?
- ボタンを取り入れたデザインを考えて、プログラミングしてみよう。



## チャレンジしよう!

センサーを取り入れたデザインのモデルを作れるかな?

# SPIKE™ プライムで作ることわざモデル

英語には「The early bird catches the worm. (早起きの鳥が虫をつかまえる)」や、「Don't put all your eggs in one basket. (たまごを全部、ひとつのカゴに入れるな)」といったことわざがある。SPIKE プライムを使って、ことわざをモデルにして動かしてみよう。自分の知っていることわざを1つ選んで、作ってみよう。



## エンジニアのつもりで考えよう:

モーターや歯車をどんなふうに使えば、ことわざを動きで表せるだろう?

## 文筆家のつもりで考えよう:

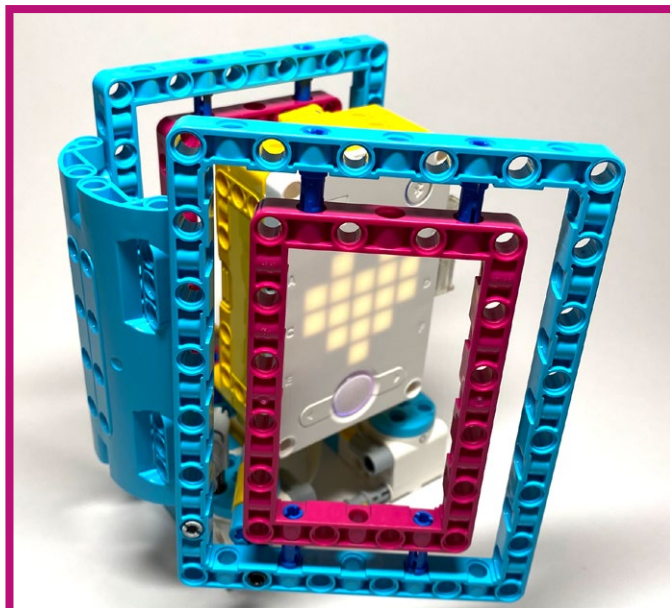
これまでの自分の経験に当てはまることわざをさがしてみよう。



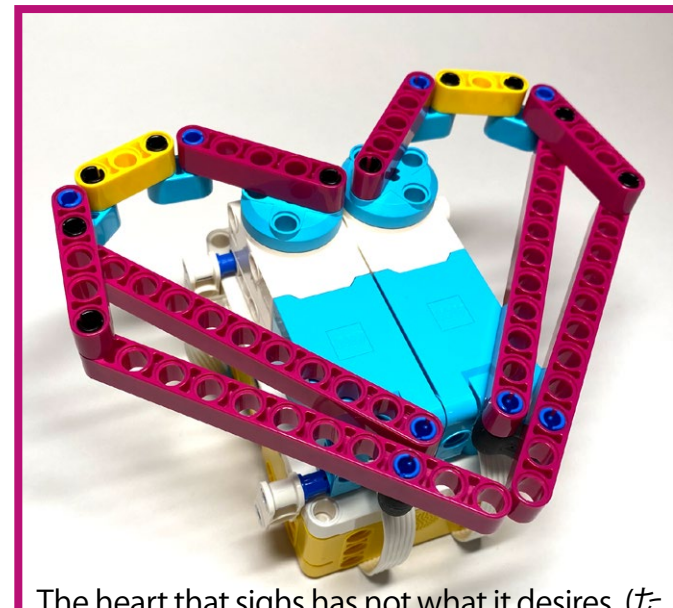
## サンプル



The early bird catches the worm. (早起きの鳥が虫をつかまえる): アメリカのことわざ



Don't judge a book by its cover. (本を表紙で判断するな): アメリカのことわざ



The heart that sighs has not what it desires. (ため息をつく心には望むものがない): フランスのことわざ

カードをうら返してくわしく見てみよう!







## 組み立てよう!

まずパーツをよく観察しよう。どんなふうに使えば、選んだことわざを表せるだろう?



## プログラミングしよう!

本を開いたりとじたりするプログラム →



```
1 from hub import port, light_matrix
2 import runloop, motor
3
4 async def main():
5     for i in range(10):
6         light_matrix.show_image(3)
7         await runloop.sleep_ms(2000)
8
9         await motor.run_for_degrees(port.A, int(0.15 * 360), 100)
10
11        light_matrix.show_image(1)
12        await runloop.sleep_ms(2000)
13
14        await motor.run_for_degrees(port.A, int(-0.15 * 360), 100)
15        light_matrix.show_image(3)
16
17 runloop.run(main())
```

## アレンジしよう:

- ことわざを表すのに、ハブについているライトを使える?
- くりかえしのループ ("repeat") や条件文 ("if-then") を使ったプログラミングを試みよう。さらに、複数のループを使えるかな?



## チャレンジしよう!

SPIKE™ プライムで作ることわざモデルを、なにかに反応して動作するインタラクティブモデルにできる?



# おそうじマシン

レゴファンはだれもが体験済み…。よくかたづけつつも、いつもブロックやパーツがゆかに落ちてるんだ。おそうじマシンを作って、ゆかに散らばったレゴブロックや小さなレゴピースをかたづけよう。ほうき、モップ、そうじ機…ゆかがきれいになる道具なら、どんなモデルでもいいよ。



## エンジニアのつもりで考えよう:

かたづけたいスペースの広さに応じて、道具を改良する方法を考えよう。

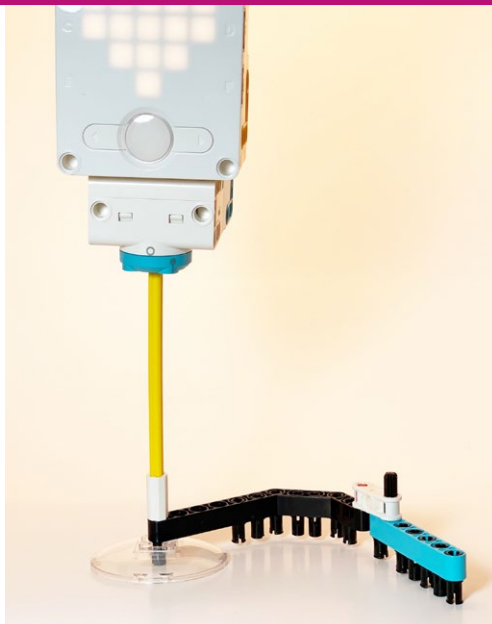
## 起業家のつもりで考えよう:

レゴファンが「ほしい!」と思うような道具を作れるかな?

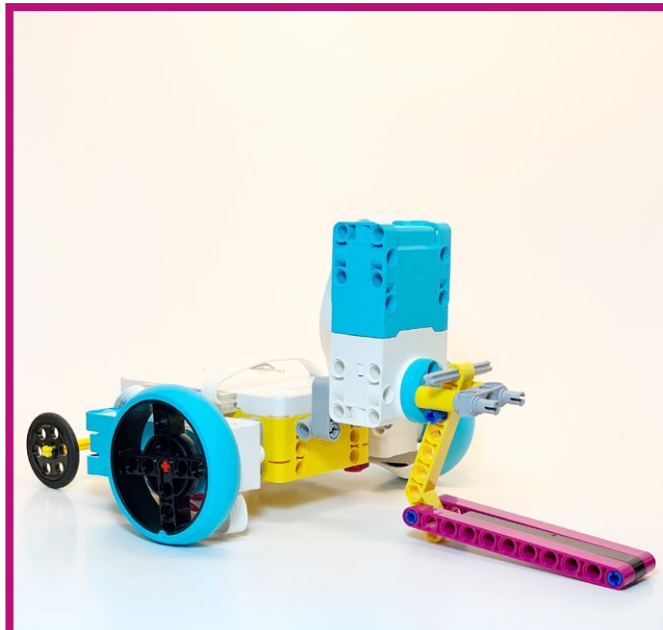


## サンプル

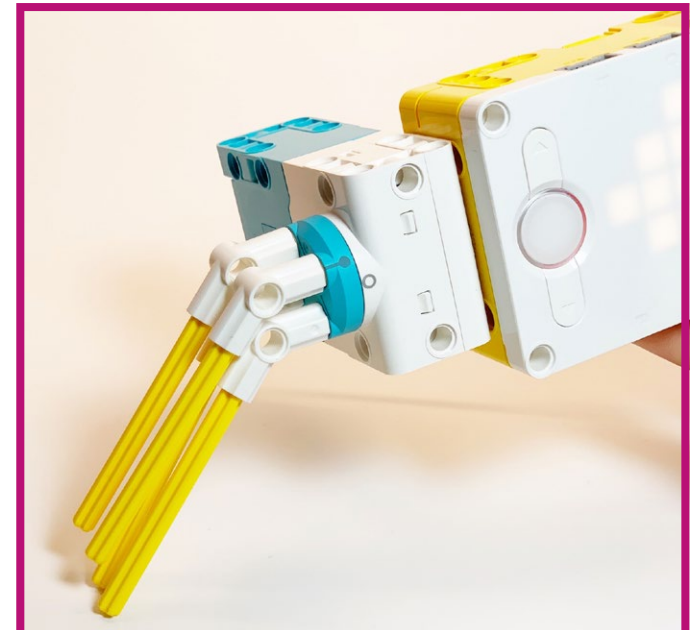
手で持って使う道具?それとも自動でおそうじしてくれる道具?どっちを作る?



円形ほうき



自走式ほうき



電動ほうき

カードをうら返してくわしく見てみよう!



## 組み立てよう!

マシンがたおれないようにするのが大切だ。

安定した重心点や支点を作って、おそうじマシンが動き回りやすいようにしてみよう。

①



②



③



## プログラミングしよう!

円形ほうきのプログラム:



```
1 from hub import port, light_matrix
2 import runloop, motor
3
4 async def main():
5     await light_matrix.write("Beginning to Clean!", 100, 500)
6
7     while True:
8         await motor.run_for_time(port.A, 2000, 100)
9         await motor.run_for_time(port.A, 2000, -100)
10        await light_matrix.write("Still Cleaning!", 100, 500)
11
12 runloop.run(main())
```

## アレンジしよう

- ボタンをおしたら動き出すようにできるかな?
- おそうじしながらお話してくれるロボットを作ってみよう。
- モーターの速度を変えると、そうじの出来ばえが変わる?
- そうじが終わったらハブの音や動作でお知らせするようにできる?



## チャレンジしよう!

ブロックやパーツをはくだけじゃなくて、拾い集めてくれるロボットを作れるかな?

# ピッチングマシン

小さなプラスチックのボールをできるだけ遠くに投げられるマシンを作ろう。



## アスリートのつもりで考えよう:

ボールをどんなふう to 投げるのか考えよう。それをもとにして、ロボットを組み立てられるかな？

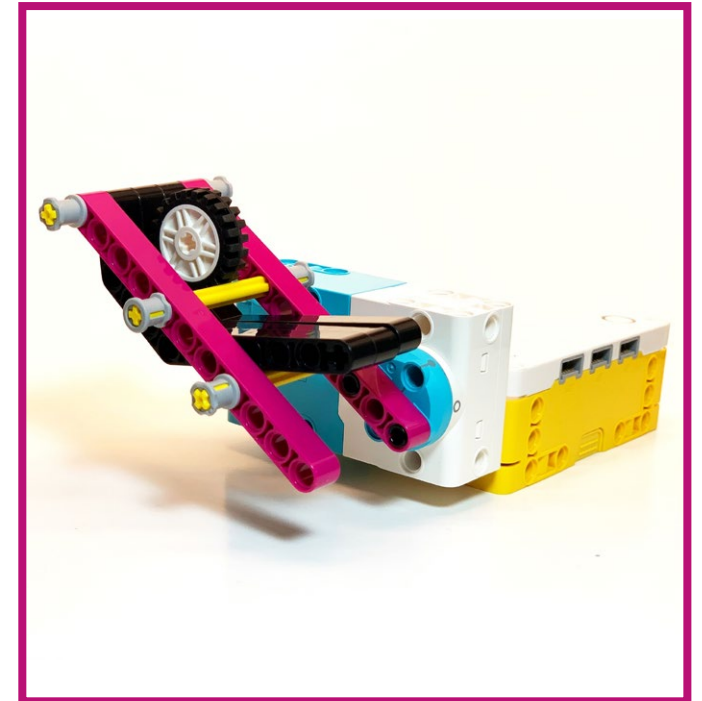
## 物理学者のつもりで考えよう:

てこの原理を利用して、ボールをもっと遠くに投げられるようにできる？



## サンプル

カタパルトやパチンコなど、実在する投石器をお手本にしてみよう。



カードをうら返してくわしく見てみよう!





## 組み立てよう!

「手首」に関節がある人間のうでをお手本にしたモデルだ!



## プログラミングしよう!

ボールを投げるうでのプログラム



```

1 from hub import port
2 import runloop, motor
3
4 async def main():
5     while True:
6         motor_A = port.A
7         await runloop.sleep_ms(2000)
8         await motor.run_to_absolute_position(port.A, 260, 300, direction=motor.SHORTEST_PATH)
9         await runloop.sleep_ms(1000)
10        await motor.run_for_degrees(motor_A, 72, 300)
11        await motor.run_to_absolute_position(motor_A, 260, 300, direction=motor.SHORTEST_PATH)
12
13 runloop.run(main())

```

## アレンジしてみよう!

- うでを動かす速さを変えたらどうなるだろう?
- 回転数を増やす/減らすとどうなるだろう?
- 最初のうでの位置を変えると、ボールが飛びきよりは変わるかな?



## チャレンジしよう!

自分の身長と同じ遠さまでボールを投げられるロボットを作ってみよう。