

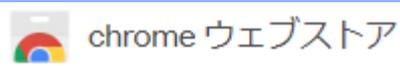
# mBotを使ってみよう

## Chromebook 編



柏市立小中学校版

# ドライバー mlink を入れる



- ①chrome ウェブストアを開く（柏市教育委員会向け）
- ②「mLink」を選び「chromeに追加」をタップ
- ③「mLinkをインストールしました」メッセージが表示され完了



# 内容物の確認



箱の中にある内容物を確認しましょう

## 【貸出 基本セット】



本体（電池BOX付）



マニュアル・トレースマップ・リモコン

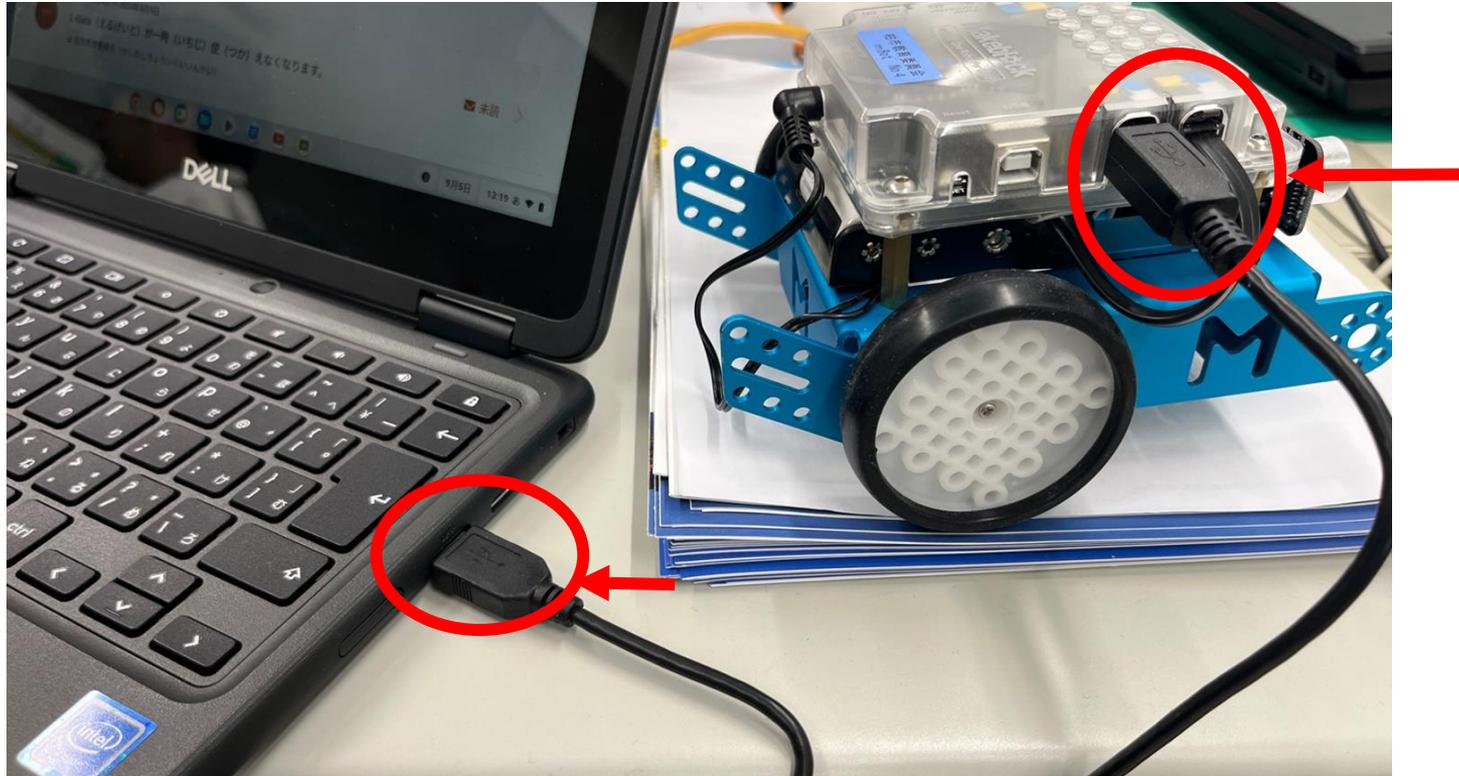


USBケーブル

※電池BOX・トレースマップ・USBケーブルの紛失が多く、不揃いになってきています  
返却の際に、入っていたものが揃っているかをご確認ください

# U S Bケーブルで接続する

Chromebook と mBot を 接続する



# 電池を入れ、電源を入れる

## ①電池BOX に単三電池 4本を入れる

※電池BOX は マジックテープで固定されているので、ずらすとはずれる

※ケーブルでつないでいるときはケーブルで給電される

## ②電源スイッチをオンにする

ブザーがなる

※音がならない時は、  
電池残量が少ない可能性があります



# ChromeBook 用の mBlock にアクセスする

- 小学校学習メニューの「リテラシー」→「プログラミング」→[mBot\(ChromeBook\)](#)
- 中学校学習メニューの「技術科の学習」→「プログラミング」→[mBot\(ChromeBook\)](#)



**リテラシー**

◆プログラミング

- スクラッチ (Scratch)
- スクラッチ (Scratchつくる)
- ポケモンプログラミング
- Why プログラミング (NHK for School)
- テキシコ (NHK for School)
- ワイワイ プログラミング (NHK)
- 動画でわかるスクラッチコマンド (NHK)
- Viscuit (ビスケット)
- Hour Of Code

[mBot\(ChromeBook\)](#)



**技術科の学習**

技術

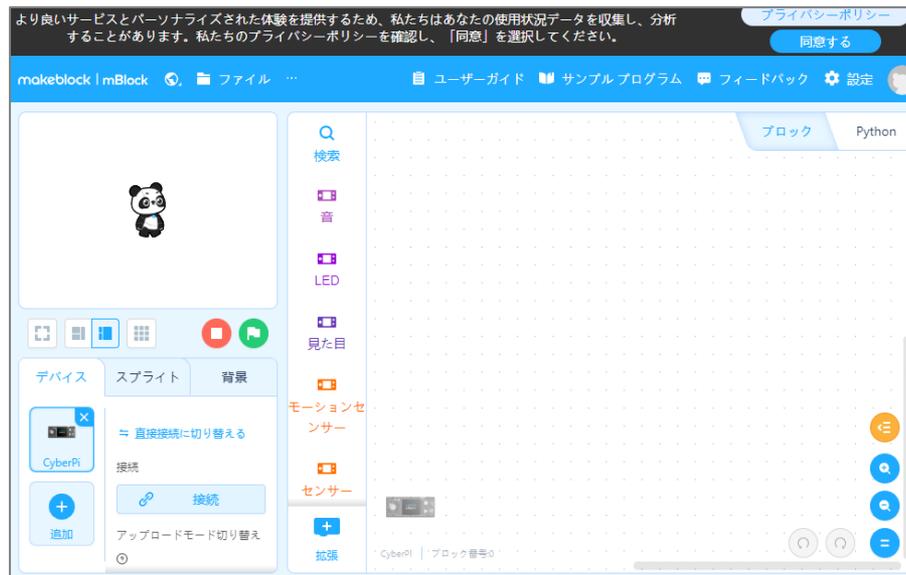
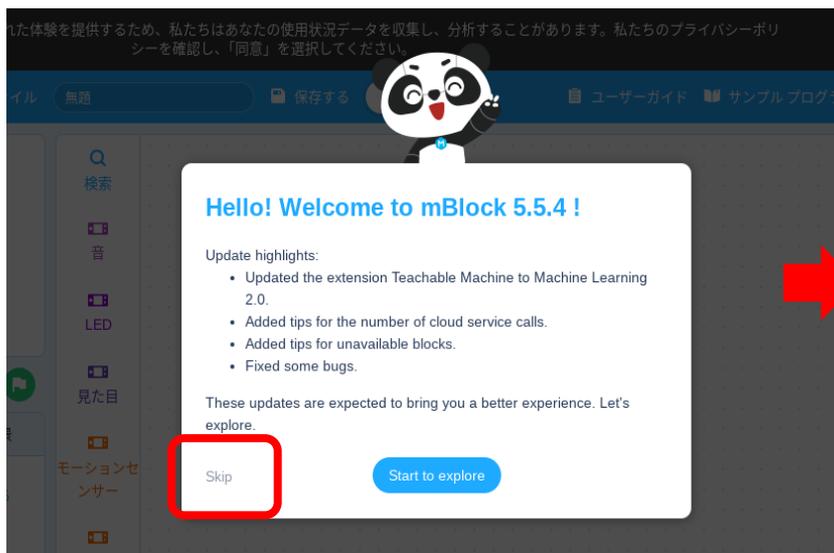
◆プログラミング

- スクラッチ (Chrome版)
- Why プログラミング(NHK for School)
- ワイワイ プログラミング (NHK)
- 動画でわかるスクラッチコマンド (NHK)
- プログラミング テキシコー (NHK)
- Viscuit (ビスケット)
- Hour of Code (プログラミング)

[mBot\(ChromeBook\)](#)

# mBlock のアプリをひらく

英文の説明がでたら「Skip」をタップする（はじめての起動時）  
→mBlockの画面が表示される



# どんなことができるの

## 4 mBotを動かそう

走る！

光る！

音が鳴る！

各種センサー搭載！



が押されたとき

前向きに 50 %の速さで 3 秒動かす

動きを止める



# どんなことができるの（動かす）

## 4 mBotを動かそう 動きのブロック

前向きに 50 %の速さで 1 秒動かす

速さと時間を決める

左向きに 50 %の速さで 1 秒動かす

曲がる 回転する時間を決める

前 ▼ 向きに 50 %の速さで動かす

動き続ける

動きを止める

※同じ動きを再現しない事が多いので注意。（電池残量・路面との摩擦・個体差 等）  
※まっすぐ走ることは難しい。  
※回転する角度を指定することはできない。回転する角度は速さと時間で調整するので、直角に曲がることは難しい。



# どんなことができるの (ライトを光らせる)

## 4 mBotを動かそう ライトのブロック

何色で何秒間光らせるかを定める

ボード上の 全て ▼ のLEDを ● 色で 1 秒点灯する



ボード上の 全て ▼ のLEDを ● 色で点灯する

光り続ける



# どんなことができるの（音をならす）

## 4 mBotを動かそう ブザーのブロック

 C4 ▼ の音階を 0.25 秒鳴らす

- ✓ C4
- D4
- E4
- F4
- G4
- A4
- B4
- C5
- D5
- E5
- F5
- G5

C4	ド
D4	レ
E4	ミ
F4	ファ
G4	ソ
A4	ラ
B4	シ

音の高さと長さを決める



# どんなことができるの (センサー)

## 4 mBotを動かそう センサーのブロック

③数値が表示される

②センサーに

①アップロードモードをオフにする

白=3 左白=2 右白=1 黒=0

照度を測定する

左右のセンサーで距離を測定する

白を判定する

# どんなことができるの (制御)

## 4 mBotを動かそう 制御のブロック

数値は必ず半角で入力する

1 秒待つ

もし  なら

 まで待つ

10 回繰り返す

もし  なら

 まで繰り返す

ずっと

でなければ

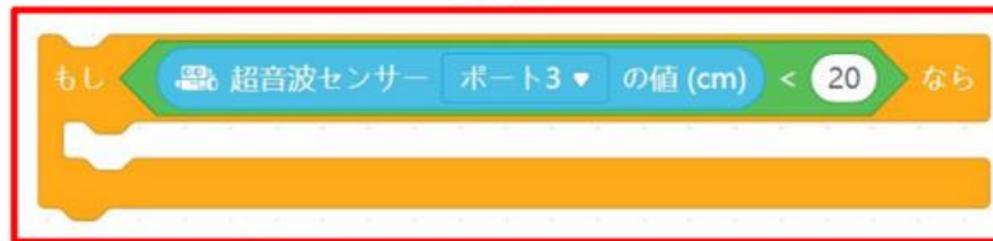
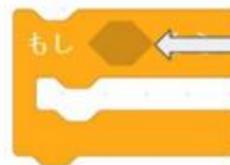
制御のブロック  
ほかの命令ブロックと組み合わせて使う



# どんなことができるの (演算)

## 4 mBotを動かそう 演算のブロック

演算のブロック  
ほかの命令ブロックと組み合わせて使う



組み合わせると「もし超音波センサーの値が20未満なら」のプログラムになる

# 初期設定をする（mBotを追加する）

- ①「デバイス」の「追加」ボタンをタップする
- ②ライブラリの一覧から「mBot」を選択し、「OK」する



# 初期設定（ファームのアップデート）

mBotに残ったプログラムをリセットします

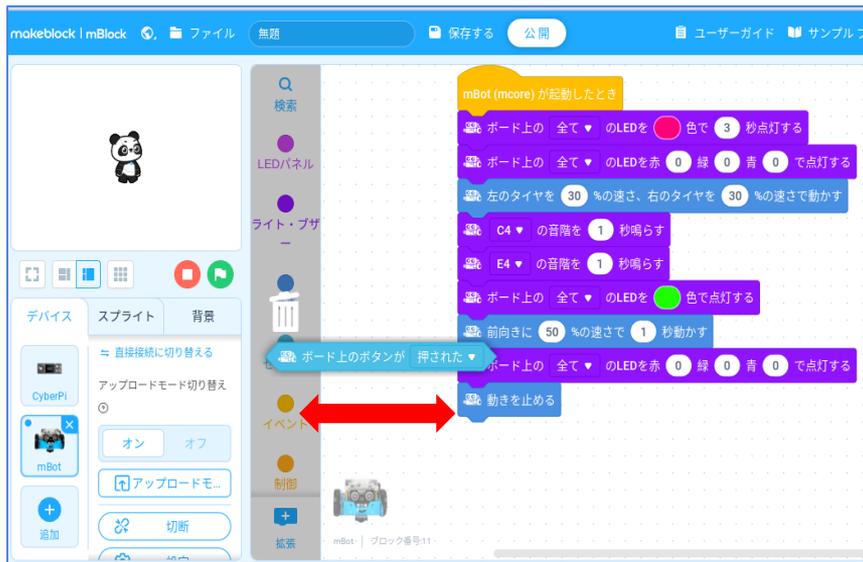
- ①画面左下の「**接続**」ボタンをタップ
- ②USB接続で表示された画面でを入れて「**接続**」をタップ  
※「**接続**」ボタンが押せないときは USBケーブルを一度抜いてさす
- ③「**設定**」ボタンをタップ
- ④「**ファームウェアを更新する**」をタップし、「オンラインのファームウェアを更新する」メッセージ画面で「**アップデート**」ボタンをタップする（※アップデート画面が消えるまで そのまま待つ）



- ⑤一度接続がきれるので、再度「**接続**」をタップ

# プログラムの作成方法

- 命令ブロックを移動して組み合わせる
- 命令ブロックを消すときは、道具の並んでいるところへドラッグしてゴミ箱へ
- 2本指タップ(右クリック)でメニュー表示 (複製・削除など)



# 3つのセンサーと値の確認

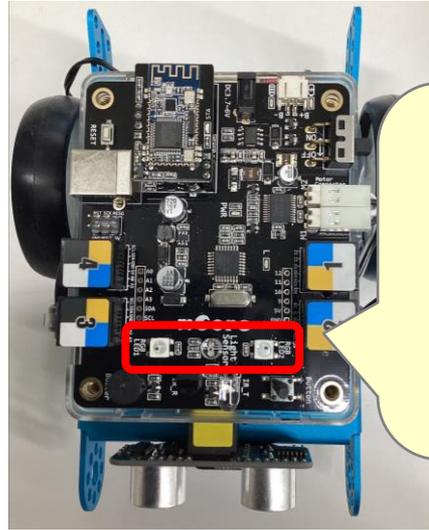
- ①センサーをクリック
- ②チェックボックスにチェックを入れる
- ③値が表示される

※画面を大きくして  
数字の変化を確認  
しましょう

The screenshot shows the mBlock software interface. At the top, there's a blue header with 'makeblock | mBlock', a search bar, and buttons for '保存する' (Save) and '公開' (Publish). Below the header, there's a search bar and a list of sensors. Three sensors are selected with checkmarks and highlighted by a red box: '光センサー' (Light Sensor), '超音波センサー' (Ultrasonic Sensor), and 'ライトレースセンサー' (Light Race Sensor). A red arrow points from the 'センサー' (Sensors) button in the left sidebar to the selected sensors. Another red arrow points from the '確認画面が大きくなります' (Confirmation screen will be larger) text box to the '確認画面' (Confirmation screen) button in the bottom left. The main workspace shows a script with several blocks, including 'ボード上のボタンが押された' (Board button pressed), '赤外線リモコン A ボタンが押された' (Infrared remote control A button pressed), and '赤外線メッセージ hello を送る' (Send infrared message hello). A small mBot robot is visible in the bottom right corner.

# 光センサーと値の確認

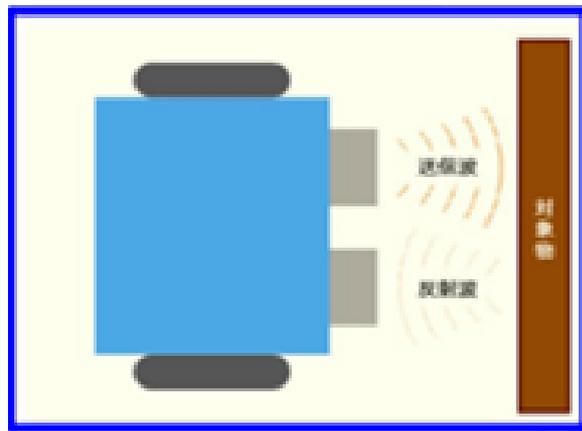
光センサーの部分を  
手でかくすと暗くなるよ



 mcore: 光センサー ボード上の光センサーの値 998

暗くなると数字が小さくなり、明るくなると数字が大きくなります (0 ~ 1023)

# 超音波センサーと値の確認

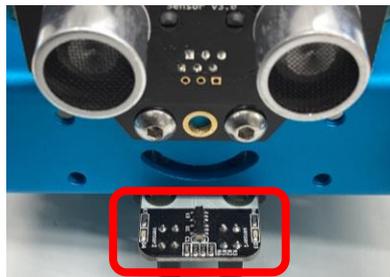


超音波センサーの前に  
対象物（手やノート等）を  
おいてみよう

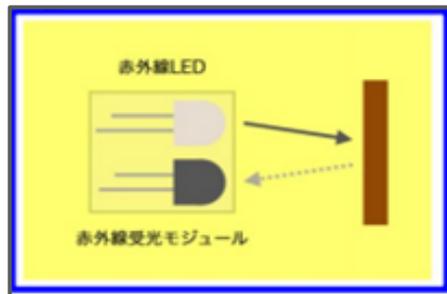
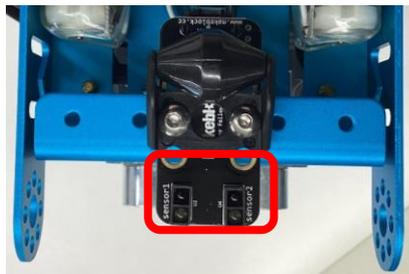
📱 mcore: 超音波センサーポート3の値 (cm) 400

対象物までの距離が表示されます（3～400cm）

# ライトレースセンサーと値の確認



裏側



LEDから発光された赤外線が、白い線や、黒い線にあたり、反射してくる光の量によって、白か黒かを見分けます

🚗 mcore: ライトレースセンサー ポート2 の値

3

センサーの値	判定の種類
0	両方のセンサーが黒
1	左のセンサーが黒、右のセンサーが白
2	左のセンサーが白、右のセンサーが黒
3	両方のセンサーが白



左右のセンサーが白黒どちらの上にあるかで値が変わります

# アップロードモードについて



## 「アップロードモード切り替え」を**オン**の場合 (ケーブルなしで動く)

→ プログラム作成後、mBotとケーブルを接続し送る作業が毎回必要  
再度アップロードする前に、**必ず**  
**ファームウェアアップデート**をしましょう

自由に動かす場合は  
線が無い方が便利



## 「アップロードモード切り替え」を**オフ**の場合 (ケーブルありで動く)

→ プログラム作成後、そのまま命令が実行できる

何度もプログラムを修正  
する場合は便利

# アップロードモードについて

「アップロードモード切り替え」を**オン**の場合  
(ケーブルなしで動く)

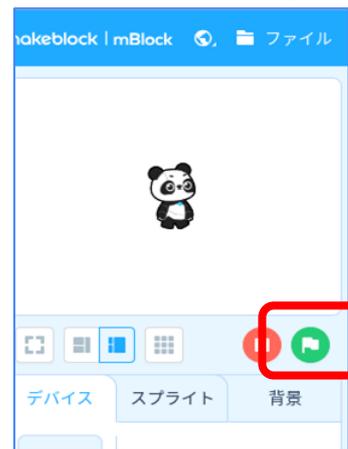
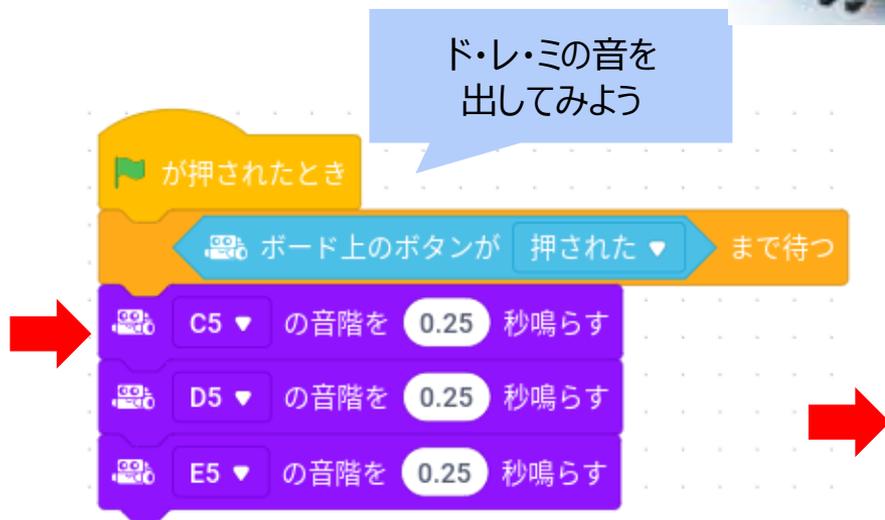
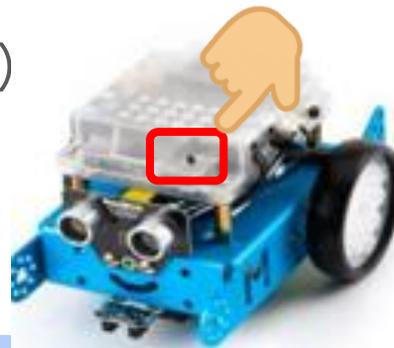
The screenshot shows the Scratch environment with the 'Upload Mode Switch' turned ON. The left sidebar shows the 'Event' category highlighted in blue. The main workspace contains a script starting with 'mBot (mcore) が起動したとき' (When mBot (mcore) starts), followed by 'ボード上のボタンが 押された ▼ とき' (When button on board is pressed), 'フラグが押されたとき' (When flag is pressed), 'スペース ▼ キーが押されたとき' (When space key is pressed), 'message ▼ を受け取ったとき' (When message received), 'message ▼ を送る' (Send message), and 'message ▼ を送って待つ' (Send message and wait).

「アップロードモード切り替え」を**オフ**の場合  
(ケーブルありで動く)

The screenshot shows the Scratch environment with the 'Upload Mode Switch' turned OFF. The left sidebar shows the 'Event' category highlighted in blue. The main workspace contains a script starting with 'mBot (mcore) が起動したとき' (When mBot (mcore) starts), followed by 'ボード上のボタンが 押された ▼ とき' (When button on board is pressed), 'フラグが押されたとき' (When flag is pressed), 'スペース ▼ キーが押されたとき' (When space key is pressed), 'message ▼ を受け取ったとき' (When message received), 'message ▼ を送る' (Send message), and 'message ▼ を送って待つ' (Send message and wait).

# 音を出してみよう（基本のプログラム）

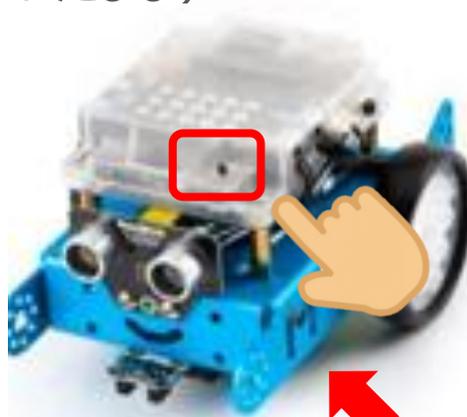
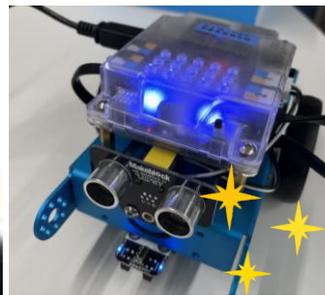
- ① アップロードモードをオフにする（USBの線はつないだまま）
- ② プログラムを作成する
- ③ 旗マーク  をタップする（プログラム実行中）
- ④ mBotのボタンを押す→音が出るか確認



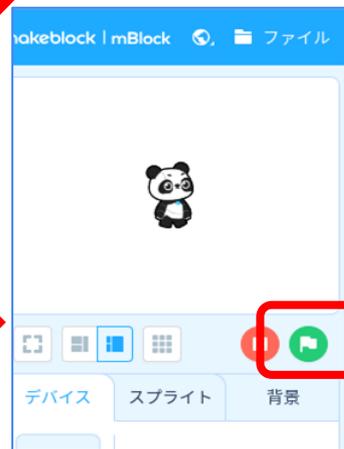
ドレミファソラシド… → CDEFGABC…になります。また、アルファベット横の数字を大きくすると高い音になります。

# ランプをつけてみよう（基本のプログラム）

- ① アップロードモードをオフにする（USBの線はつないだまま）
- ② プログラムを作成する
- ③ 旗マーク  をタップする（プログラム実行中）
- ④ mBotのボタンを押す→ランプが光るか確認

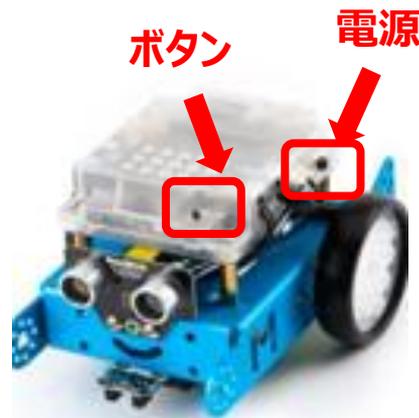


赤→黄→青  
の順番にランプを  
光らせてみよう



# 動かしてみよう（基本のプログラム）・・・アップロードモード

- ①アップロードモードを「オン」にする（USBの線はつないだまま）
- ②プログラムを作成する（※イベントの命令が変わる）
- ③「アップロード」ボタンをタップしプログラムを送ったら、mBotからUSBの線をぬく
- ④mBotのボタンを押す→車が動くか確認  
（※再度動かす時は、mBotの電源を入れ直してボタンを押す）

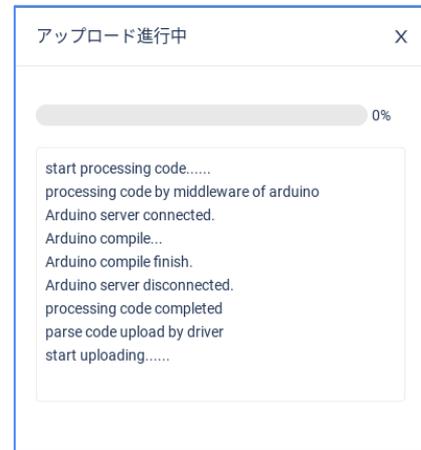
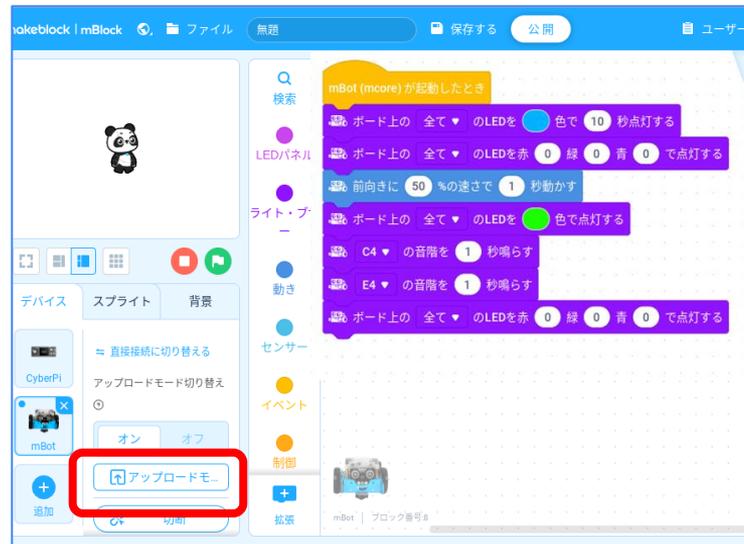


# 動かしてみよう（基本のプログラム）・・・アップロードモード

## 【プログラムを変更した時】

- ① mBotをUSBでつなぐ
- ② 「接続」ボタンをタップして接続する
- ③ **ファームウェアのアップデートをする**
- ④ 「アップロード」ボタンをタップしプログラムを送る

接続が切れるので、  
再度「接続」してから  
アップロードしましょう



# サンプルプログラム①（衝突しない車）

超音波センサー（距離を測る）

障害物に近づいたら（20cm）動きを止めるプログラムを作ろう



止めたい距離を  
入れる



障害物を置いて  
車を動かしてみよう

# サンプルプログラム② (センサーライト)

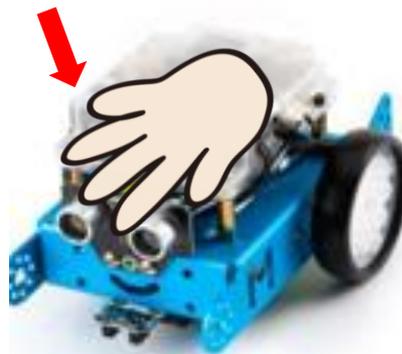
光センサー (明るさを測る)

暗くなったら点灯し、明るくなったら消すプログラムを作ろう

光センサーの部分を  
手でかくすと暗くなるよ

The code is written in a block-based programming language. It starts with a yellow block: "mBot (mcore) が起動したとき". This is followed by a blue block: "ボード上のボタンが 押された" with a dropdown arrow and "まで待つ". Below this is an orange "ずっと" (loop) block. Inside the loop is a green "もし" (if) block: "光センサー ボード上の光センサー" with a dropdown arrow, "の値 < 700" in a white circle, and "なら". Under the "もし" block are two purple blocks: "ボード上の 全て" with a dropdown arrow, "のLEDを" followed by a white circle and "色で点灯する". Below these is an orange "でなければ" (else) block, followed by another purple block: "ボード上の 全て" with a dropdown arrow, "のLEDを" followed by a black circle and "色で点灯する". The code ends with a yellow arrow block.

光センサーの数値が  
700未満だと暗い



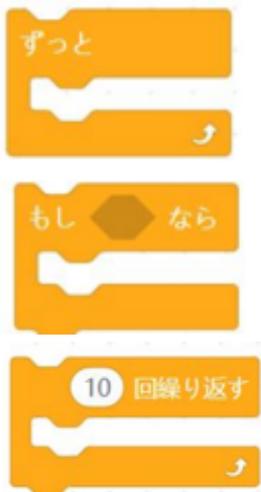
# サンプルプログラム（押しボタン信号機）

ボタンを押すと、赤から青ランプに変わり、時間がたったら青の点滅になり、赤ランプに変わる プログラムを作ろう

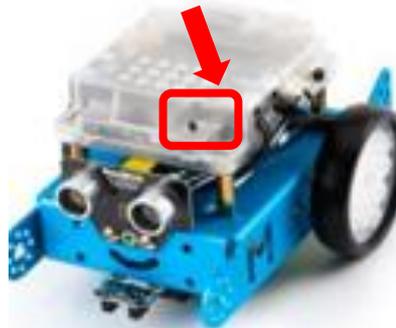
```
mBot (mcore) が起動したとき
ずっと
  ボード上の 全て のLEDを 赤 色で点灯する
  もし ボード上のボタンが 押された なら
    3 秒待つ
    ボード上の 全て のLEDを 青 色で 5 秒点灯する
    4 回繰り返す
    ボード上の 全て のLEDを 青 色で 0.5 秒点灯する
    ボード上の 全て のLEDを 黒 色で 0.5 秒点灯する
  ずっと
```

はじめは赤ランプ点灯

点滅させるには青と黒を交互に点灯



ボタン



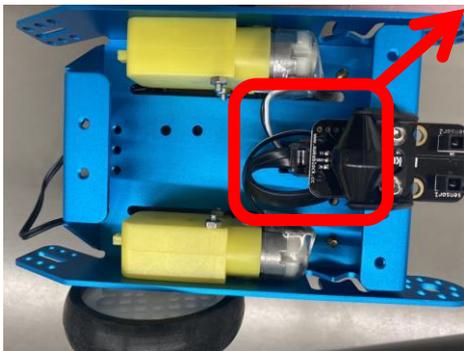
# 困った時には

【動かない時には確認しましょう】

- ・ファームウェアの更新を試みよう
- ・電池の確認をしよう
- ・センサーをさしているポートが合っているか確認



ポート2 : ライトレースセンサー  
(白黒色識別)



ポート3 : 超音波センサー  
(距離)

