

プログラミング教育研修

隣の人と話してみよう

- ①この夏の予定
- ②プログラミング教育
私は、こんなことやってます！
- ③小学校でプログラミング教育が
必修化した理由

アンプラグド プログラミング 「ヘルプミー」

- ①森で誤って毒キノコを食した あなた
- ②最後のチカラを振り絞り、家にいる
ロボットへ命令を送信
- ③ロボットは、あなたの元へ辿り着くのか…

- ・ 後ろ座席の人は、1分以内に命令を書く
- ・ 前座席の人は、命令通りに動く
- ・ 辿り着いた場所を2人で確認する

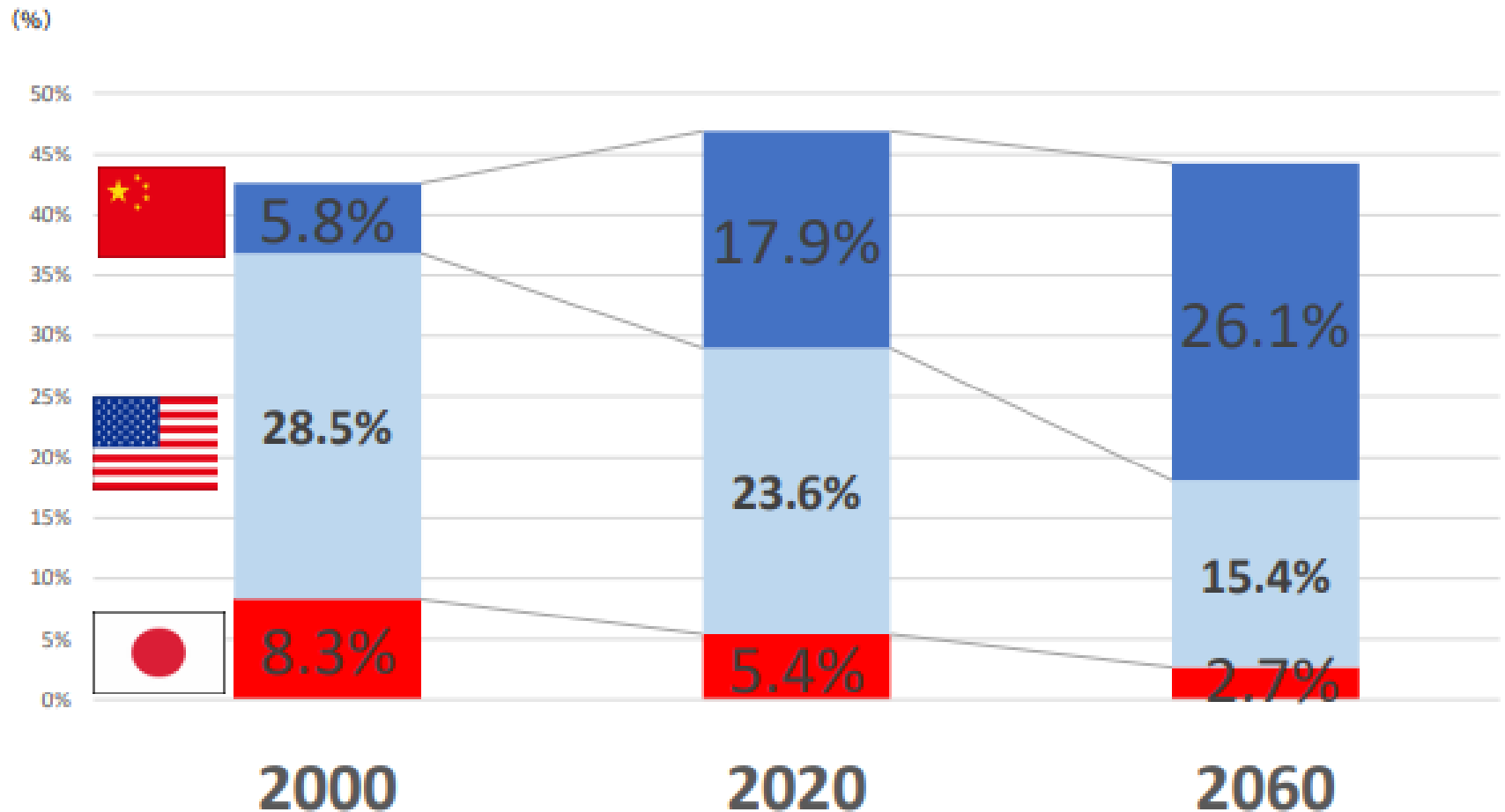
小学校でプログラミング教育を 必修化した理由

便利な機械が「魔法の箱」ではなく、**プログラミングを通じて人間の意図した処理を行わせることができるもの**であり、人間の叡智が生み出したものであることを理解できるようにすることは、時代の要請として受け止めていく必要がある。

小学校段階におけるプログラミング教育の在り方について
(議論のまとめ) 平成28年6月16日

1. グローバル化の進展①

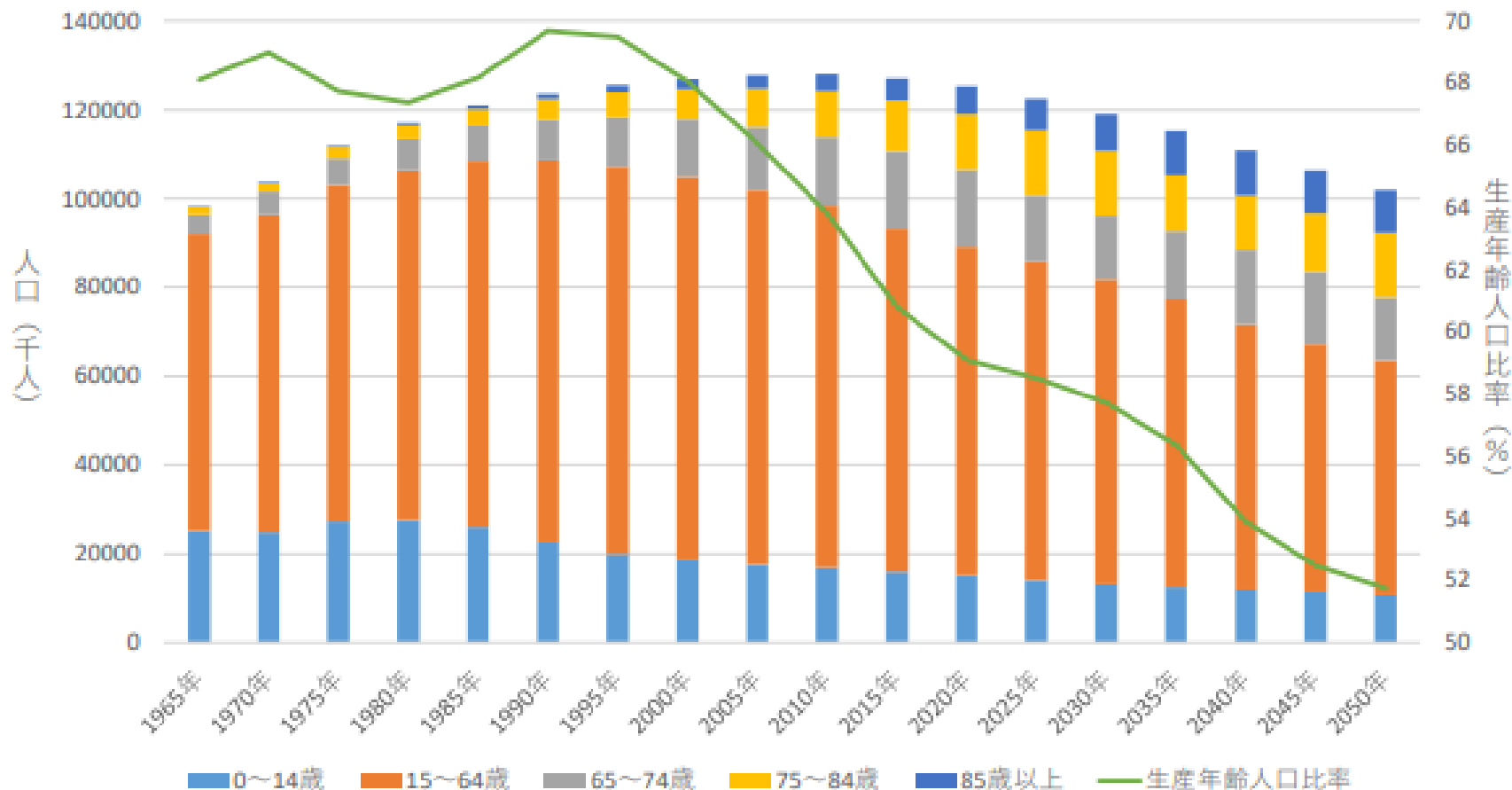
(世界GDPに占める日本の割合は大幅に低下)



(出所) World Bank「World Development Indicators」、OECD「The Long Game: Fiscal Outlooks to 2060 Underline Need for Structural Reform」より作成。

2. 人口減少・少子高齢化の進展

○2050年には日本の人口は約1億人まで減少する見込み。生産年齢人口比率は約5割に。

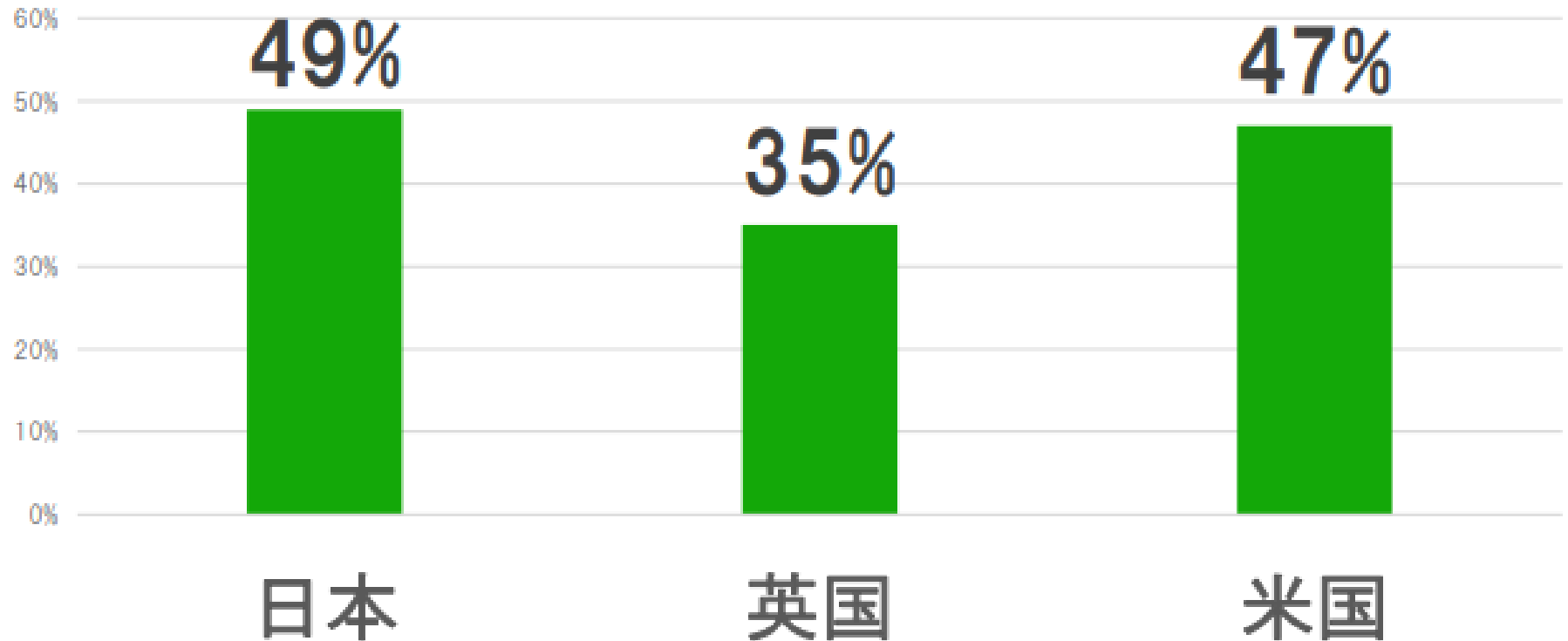


(備考) 将来推計人口は出生中位(死亡中位)。生産年齢人口は15～64歳の人口。

(出所) 国立社会保障・人口問題研究所「日本の将来推計人口(平成29年推計)」より作成。

人工知能やロボット等による代替可能性

- 10～20年後に、日本の労働人口の約49%が、技術的には人工知能やロボット等により代替できるようになる可能性が高いとの推計結果が出ている。



※日本データは国内601種類の職業について、それぞれ人工知能やロボット等で代替される確率を試算。

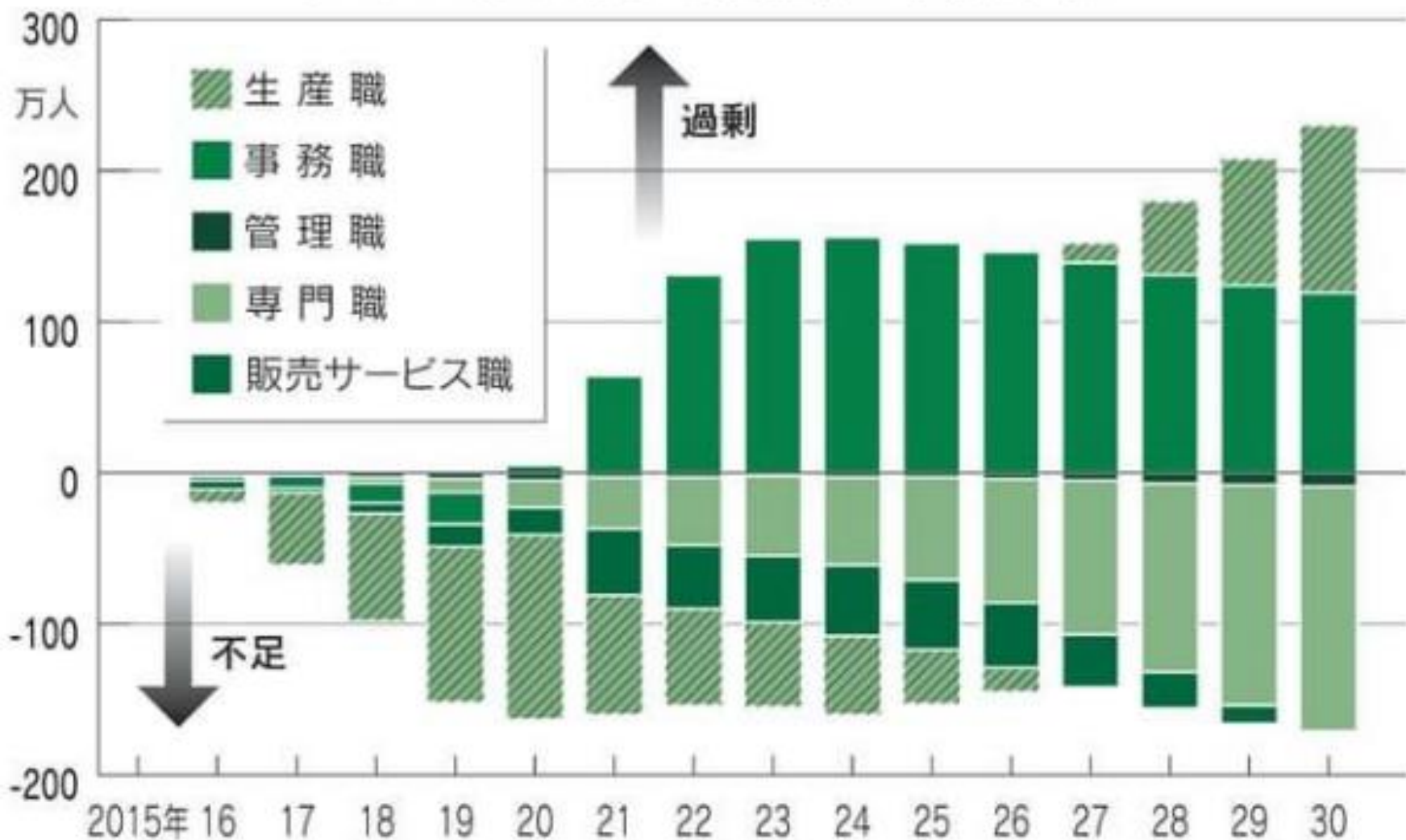
※日本データは、株式会社野村総合研究所と英オックスフォード大学のマイケル A. オズボーン准教授及びカール・ベネディクト・フレイ博士との共同研究（2015年）

米国データは、オズボーン准教授とフレイ博士の共著“The Future of Employment”（2013年）

英国データは、オズボーン准教授、フレイ博士、デロイトトーマツコンサルティング社による報告結果（2014年）から採っている。

（出典）2015年12月2日株式会社野村総合研究所News Release

デジタル化の加速で、事務職は余剰人員に



(出所)三菱総合研究所、2015年を起点に30年までの労働需給を推計

出典：日本経済新聞電子版2021年11月29日 5:00

日本のデジタル競争力は28位。 先進諸国の中では低い

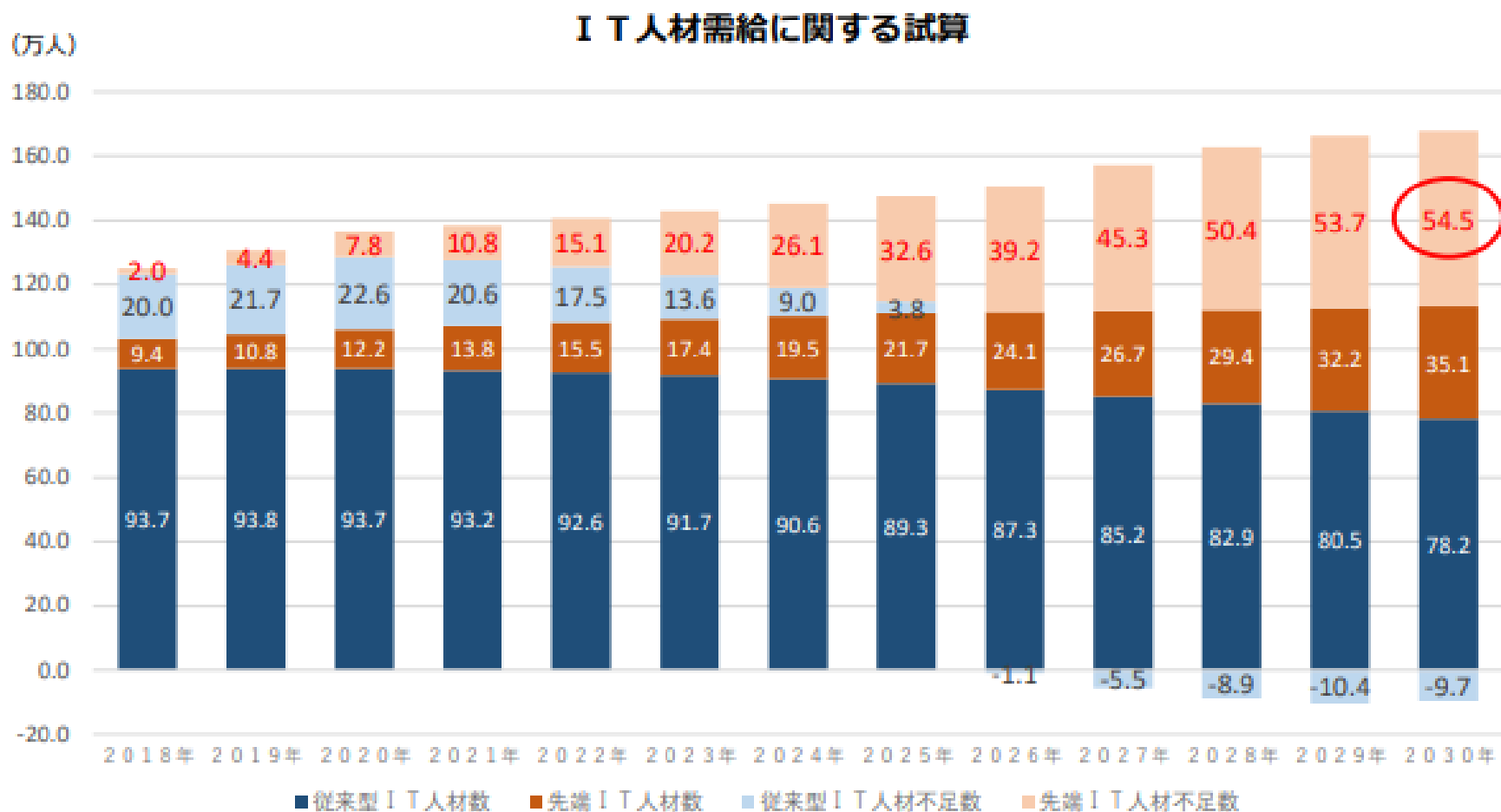
世界デジタル競争力ランキング（2021）

順位	国名	順位	国名	順位	国名
1	米国 (0)	23	ニュージーランド (↓1)	45	ハンガリー (↑2)
2	香港 (↑3)	24	フランス (0)	46	インド (↑2)
3	スウェーデン (↑1)	25	エストニア (↓4)	47	スロバキア (↑3)
4	デンマーク (↓1)	26	ベルギー (↓1)	48	トルコ (↓4)
5	シンガポール (↓3)	27	マレーシア (↓1)	49	ジョーダン (↑4)
6	スイス (0)	28	日本 (↓1)	50	ルーマニア (↓1)
7	オランダ (0)	29	カタール (↑1)	51	ブラジル (0)
8	台湾 (↑3)	30	リトアニア (↓1)	52	ブルガリア (↓7)
9	ノルウェー (0)	31	スペイン (↑2)	53	インドネシア (↑3)
10	UAE (↑4)	32	カザフスタン (↑4)	54	ウクライナ (↑4)
11	フィンランド (↓1)	33	チェコ (↑2)	55	クロアチア (↓3)
12	韓国 (↓4)	34	ポルトガル (↑3)	56	メキシコ (↓2)
13	カナダ (↓1)	35	スロベニア (↓4)	57	ペルー (↓2)
14	英国 (↓1)	36	サウジアラビア (↓2)	58	フィリピン (↓1)
15	中国 (↑1)	37	ラトビア (↑1)	59	コロンビア (↑2)
16	オーストリア (↑1)	38	タイ (↑1)	60	南アフリカ (0)
17	イスラエル (↑2)	39	チリ (↑2)	61	アルゼンチン (↓2)
18	ドイツ (0)	40	イタリア (↑2)	62	モンゴル (0)
19	アイルランド (↑1)	41	ポーランド (↓9)	63	ボツワナ
20	オーストラリア (↓5)	42	ロシア (↑1)	64	ベネズエラ (↓1)
21	アイスランド (↑2)	43	キプロス (↓3)		
22	ルクセンブルグ (↑6)	44	ギリシャ (↑2)		

（備考）括弧内は前年度との比較。

（出所）IMD「World Digital Competitiveness Ranking」(2021)より作成。

人材のスキル転換が停滞した場合、2030年には 先端 I T 人材が54.5万人不足。



(出所) 経済産業省委託調査「IT人材需給に関する調査(みずほ情報総研株式会社)」(2019年3月)より作成。

TECH KIDS GRAND PRIX 2021



◇手話と点字のアプリ

「聴覚や視覚に障がいがある人たちのことを知るにつれて、何か自分がつくったもので役に立てないかと考えるようになりました」

問題発見・解決能力

育成する資質・能力

知識及び技能

《小》身近な生活でコンピュータが活用されていることや**問題の解決には必要な手順がある**ことに気付くこと。

《中》社会におけるコンピュータの役割や影響を理解するとともに、**簡単なプログラムを作成できる**ようにすること。

思考力、判断力、表現力等

発達の段階に即して、「プログラミング的思考」（**自分が意図する一連の活動を実現するために、どのような動きの組合せが必要であり、一つ一つの動きに対応した記号を、どのように組み合わせたらいいのか、記号の組合せをどのように改善していけば、より意図した活動に近づくのか、**といったことを論理的に考えていく力）を育成すること

学びに向かう力、人間性等

発達の段階に即して、コンピュータの働きを、**よりよい人生や社会づくりに生かそうとする態度**を涵養すること。



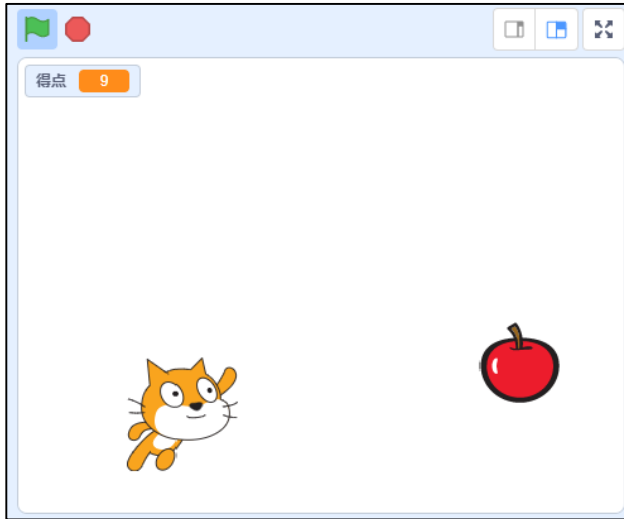
プログラミングを体験してみよう



Scratchとは

- アメリカの有名な大学、MIT（マサチューセッツ工科大学）メディアラボで開発されたビジュアルプログラミング言語です。
- ブロックを組み換えるように順序を変えたり、新しいものを足したりできるので、プログラムを書き換えるのが簡単です。作ったプログラムがうまく動かなくても、簡単にやり直せるのが特長です。

作成するプログラミング



「ネコはリンゴが大好物！」

- ・リンゴが上からランダムに落ちてきます
- ・キーボードでネコを動かしてリンゴをキャッチ！
- ・キャッチできたら+1点
地面に落ちたら-1点

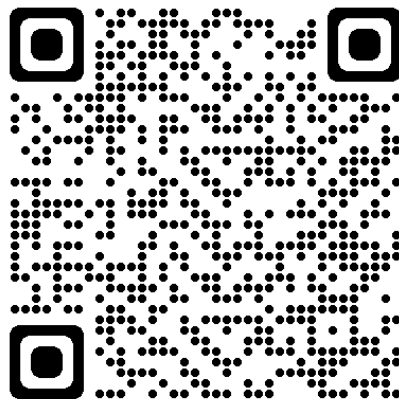
《協力》大分県次世代プログラマー発掘コンテスト
「Hello,World!」運営事務局

《目的》~~プログラミングを完成させること~~

- 必要な手順があることに気付く
- 記号の組合せをどのように改善していけば、より意図した活動に近づくのかを考える

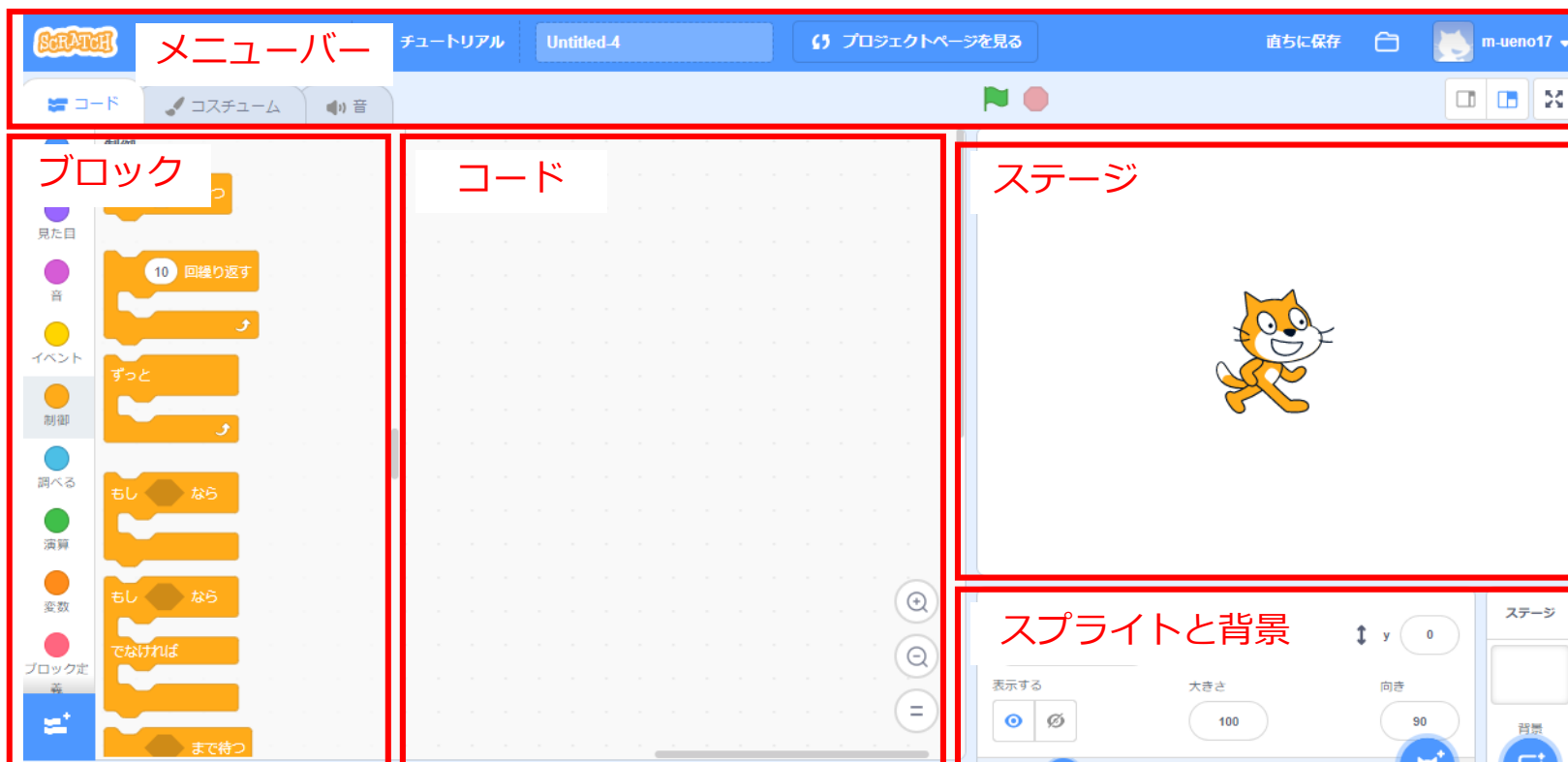
スクラッチを使う

右記サイトにアクセスし、
「作ってみよう」を
タップすれば、すぐに使えます。



<https://scratch.mit.edu/>

《作成画面》



スクラッチの基本

①色が大事！

「いつも使う旗は… そう、黄色！」
などの声かけが大事です！

②間違っても大丈夫！

違うブロックを選択しても
元の場所に持っていけば
消せます。
何度間違っても大丈夫だと
教えてあげてください。

③付け外しも簡単！

ブロックをくっつける時は
下から上！
外す時は、上から下！

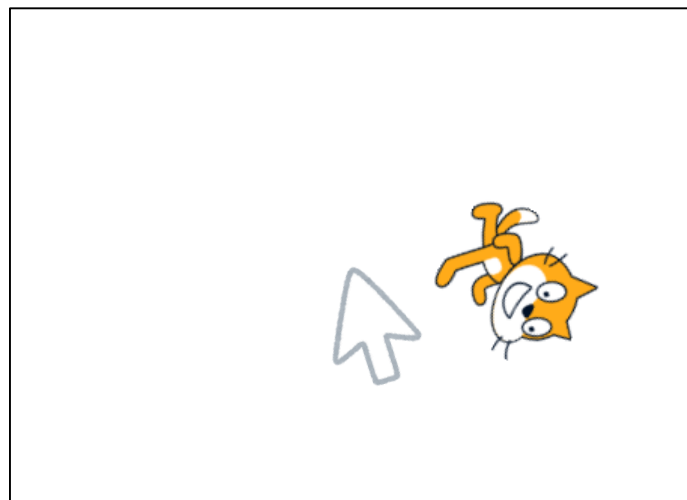




プログラミングにチャレンジ！

ネコの散歩

- ・ネコがマウスポインタを追いかけてきます。




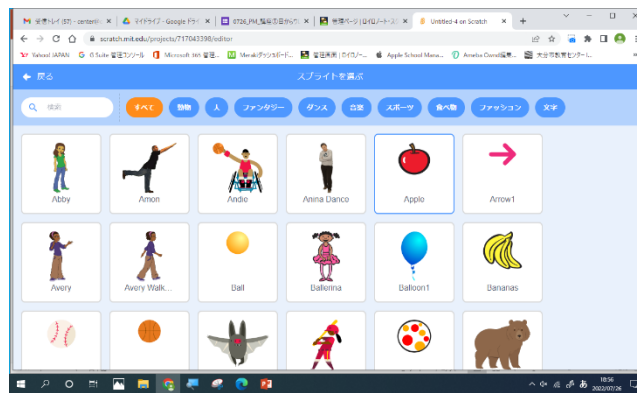
「ずっと」がないと、
どうなる？

リンゴを増やせ！①

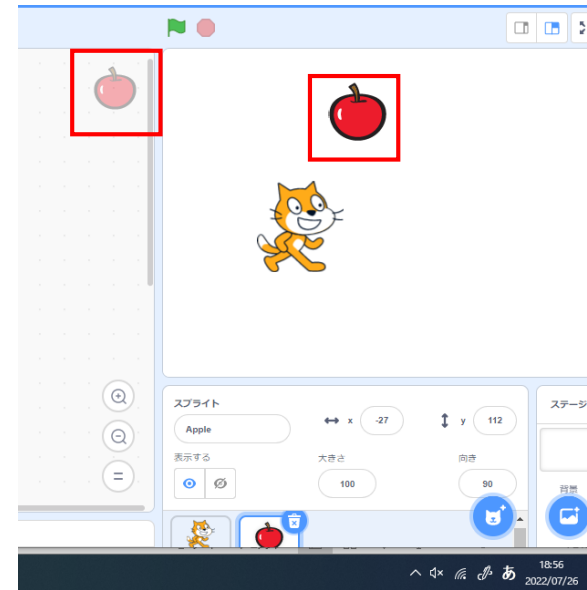
◇スプライトを「ネコ→リンゴ」に切り替えます。



①画面右下の  を
タップする



②リンゴを選択



③コードとステージ
にリンゴが出ている
ことを確認する

リンゴを増やせ！②



が押されたとき

x座標を 100 にする

ずっと

1 秒待つ

自分自身 のクローンを作る

クローンされたとき

y座標を 100 にする

x座標を -188 から 188 までの乱数 にする

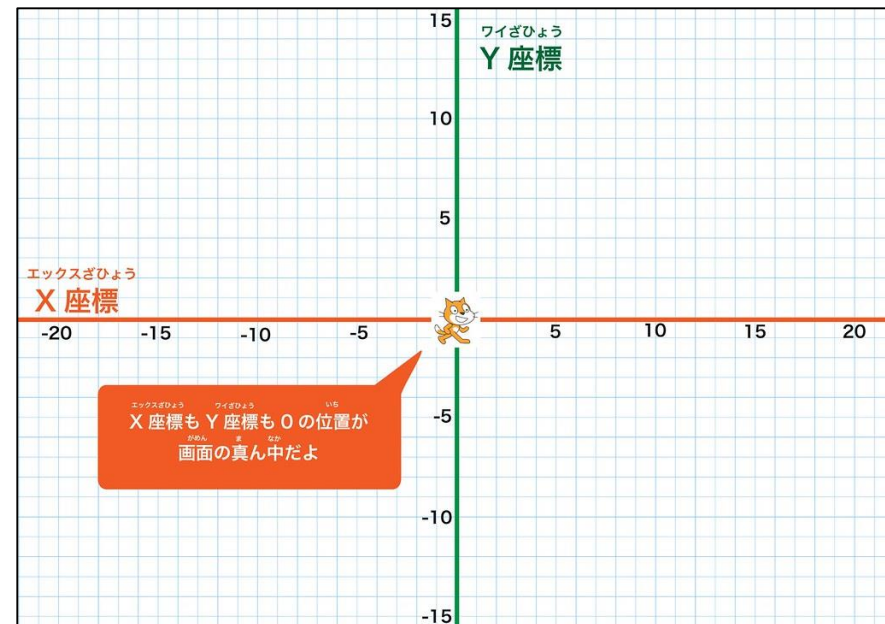
ずっと

y座標を -10 ずつ変える

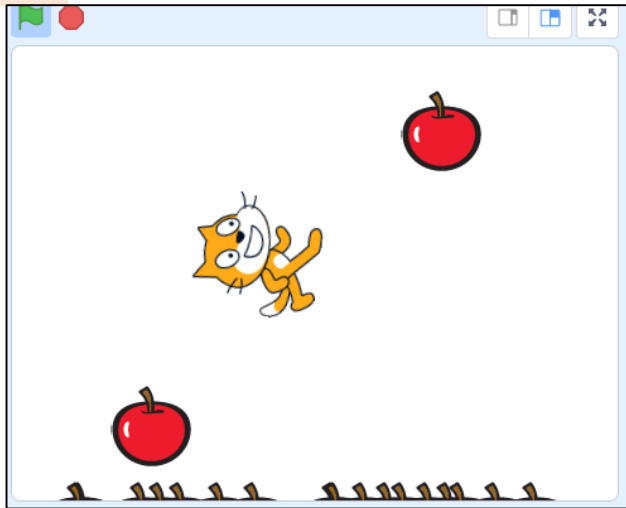
このプログラミングが
意味しているのは…？



【考え方のヒント】



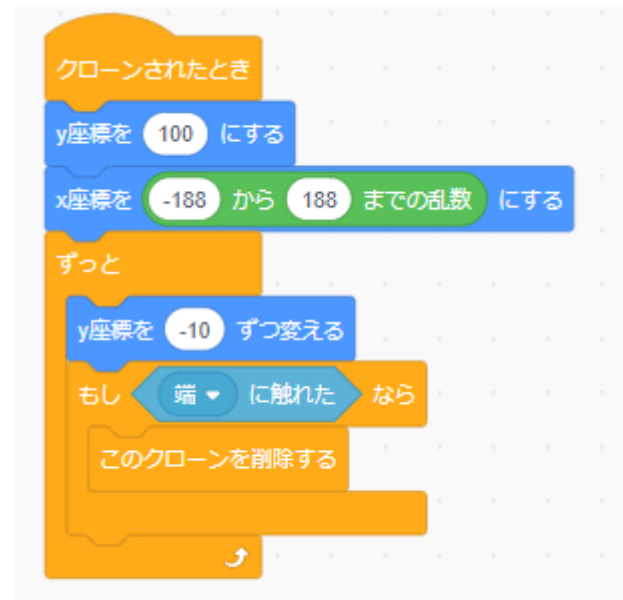
リンゴを消したい！



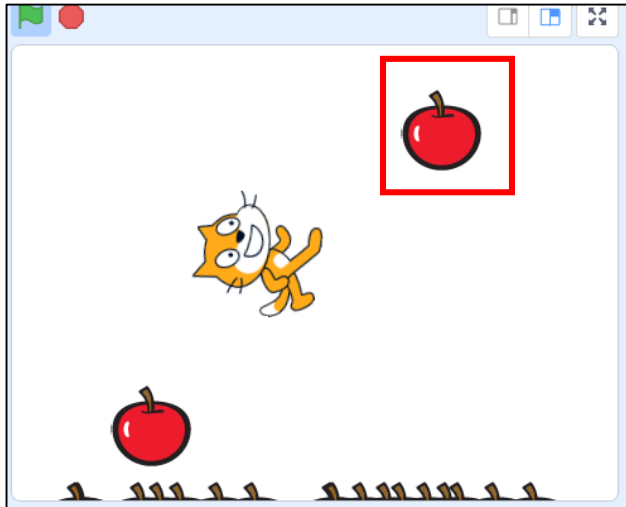
◇落ちたリンゴを消したい…



地面に落ちた
↓
「端に触れた」



宙に浮いてるリンゴ（本体）が…①



本体**だけ**見えないようにしたい！

【考え方のヒント】

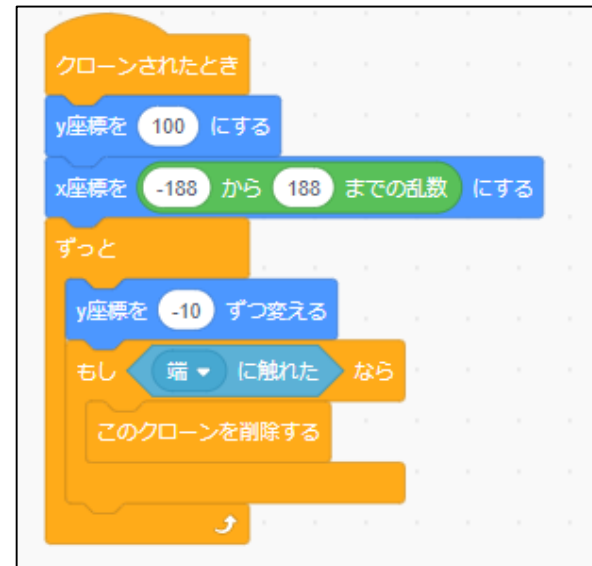


この2つのコードを
どこに入れたらいい？

《リンゴ本体》



《クローン リンゴ》



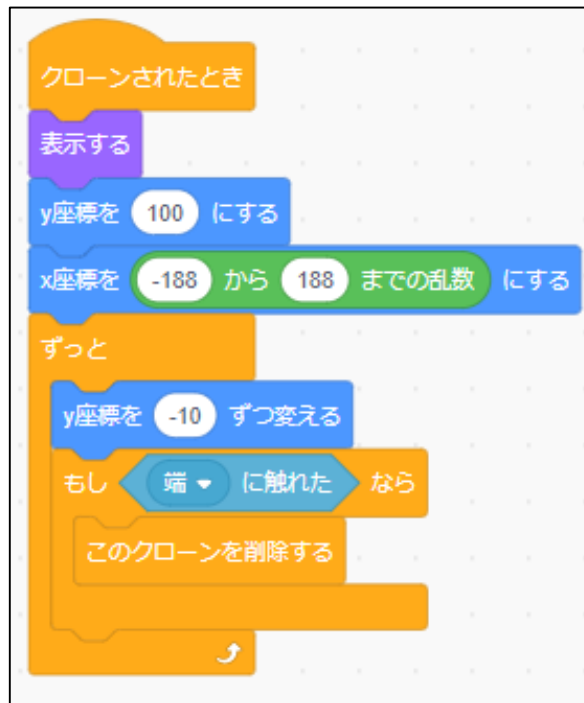
宙に浮いてるリンゴ（本体）が…②

《リンゴ本体》

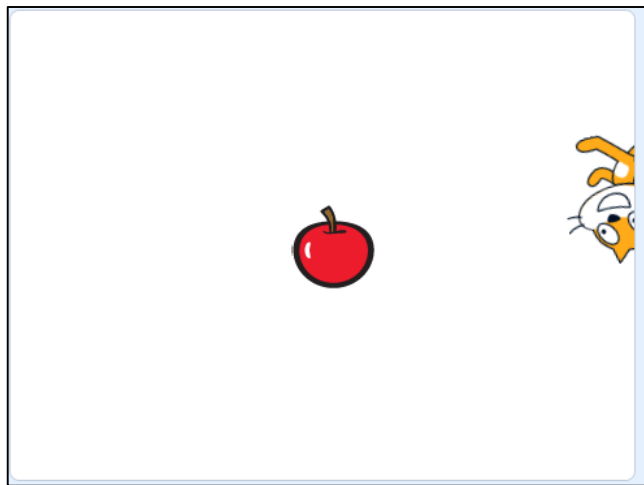


本体は「隠された状態」で
ずっと、クローンを作る

《クローンリンゴ》



クローンは「表示されて」
ずっと、落ち続ける。（y軸-10ずつ）
落ちたもの（端に触れたもの）は
削除される



リンゴが次々と落ちて
地面にも残らなくなった！

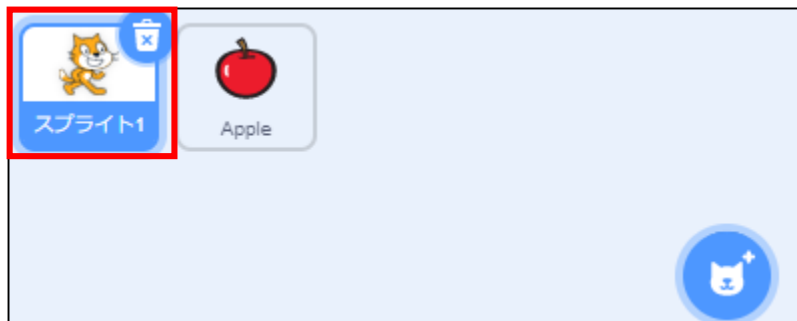
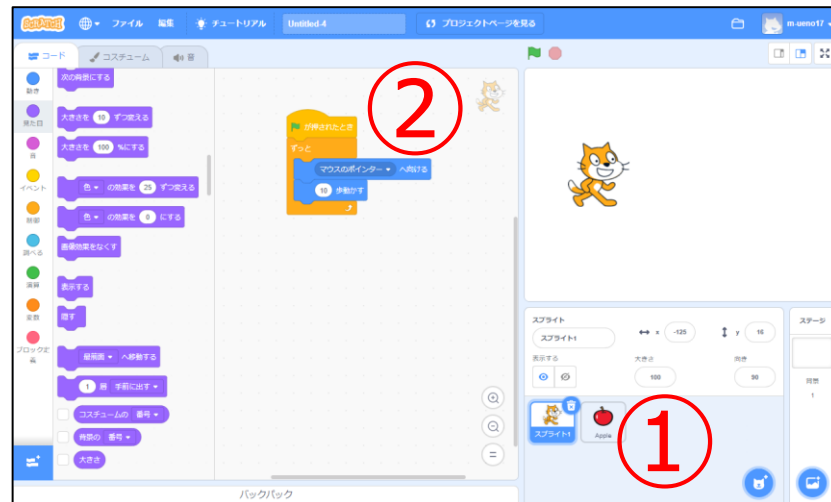


でも、ネコがいる
意味がないにや…

【Next Mission】

- ①ネコを自由に動かせるようにしよう！
- ②ネコが触ったら、リンゴが消えるようにしよう！
- ③「ネコが触ったら得点」・「地面に落ちたら減点」にしよう！

ネコの編集画面に戻る



①画面右下で、リンゴからネコに切り替える



②コード画面に、ネコが表示されていることを確認する

ネコを自由に動かす①

もし

ずっとマウスポインタを
向いて動き続ける

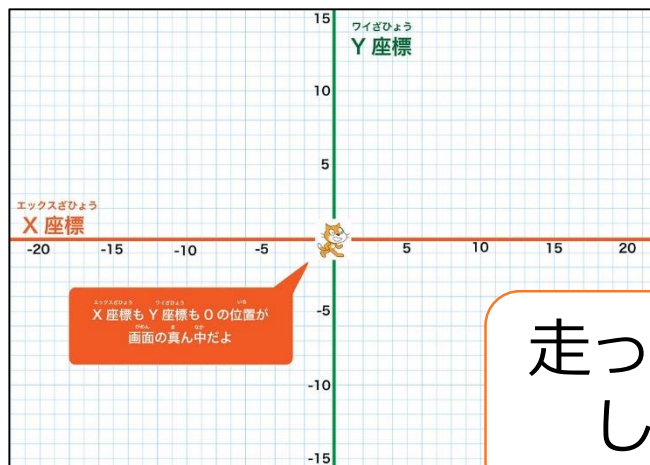
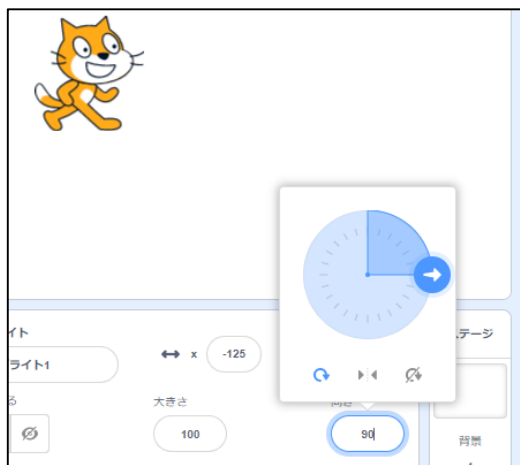
「→」を押したら右に、
「←」を押したら左に。



ネコを自由に動かす②

「→」を押したら右に

右を向いた状態で
右の方に動かしたい



走っている感じが
しないにや...



ネコを自由に動かす③

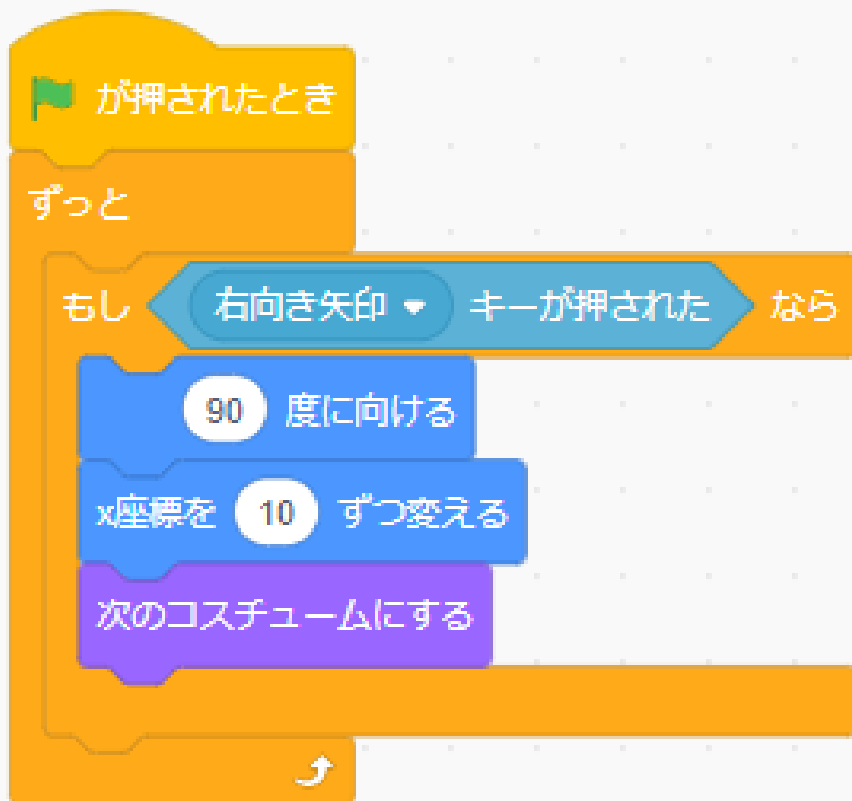


走っているように見せるには、
コスチュームを使います



繰り返すことで、走っているように見える！

ネコを自由に動かす④



《チャレンジ！》

プログラムの意味を
説明してみよう！

ネコを自由に動かす⑤



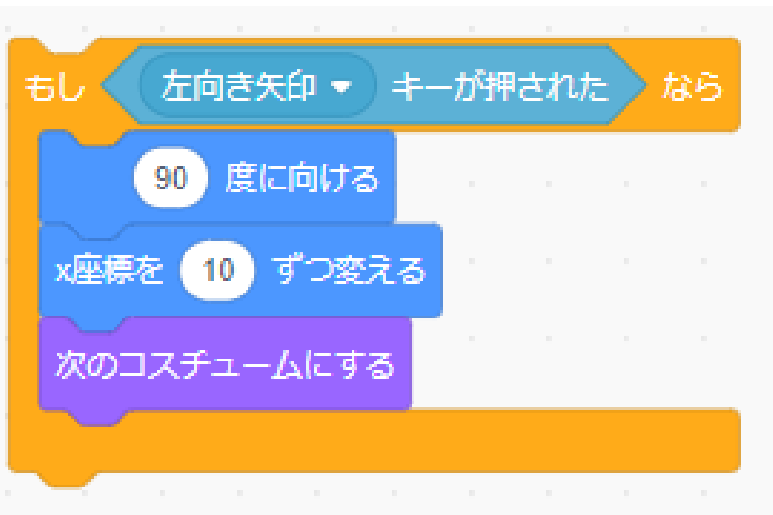
プログラムの意味を
説明してみよう！

- ① 「→」が押されたら
- ② 右側を向いて
- ③ x 軸を10移動する
- ④ コスチュームを変える
- ⑤ ①～④をずっと続ける



なるほど！
左向きも同じようにできるね

ネコを自由に動かす⑥

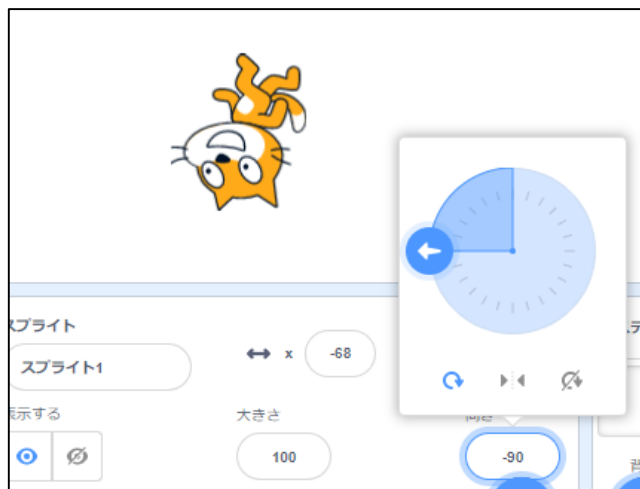


《チャレンジ！》
一カ所、修正しよう！

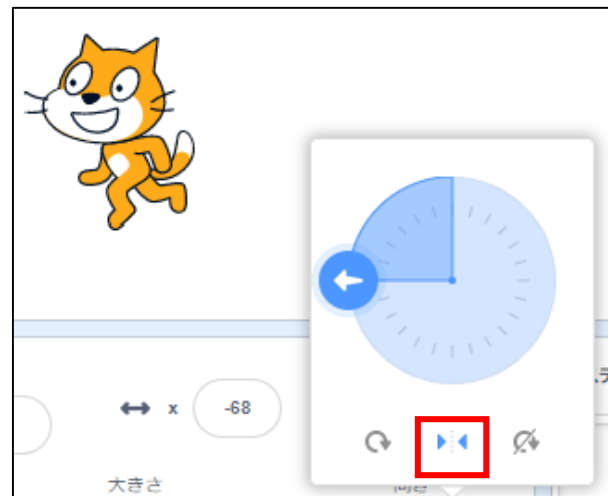
ネコを自由に動かす⑦



これじゃ
ムーンウォークにや



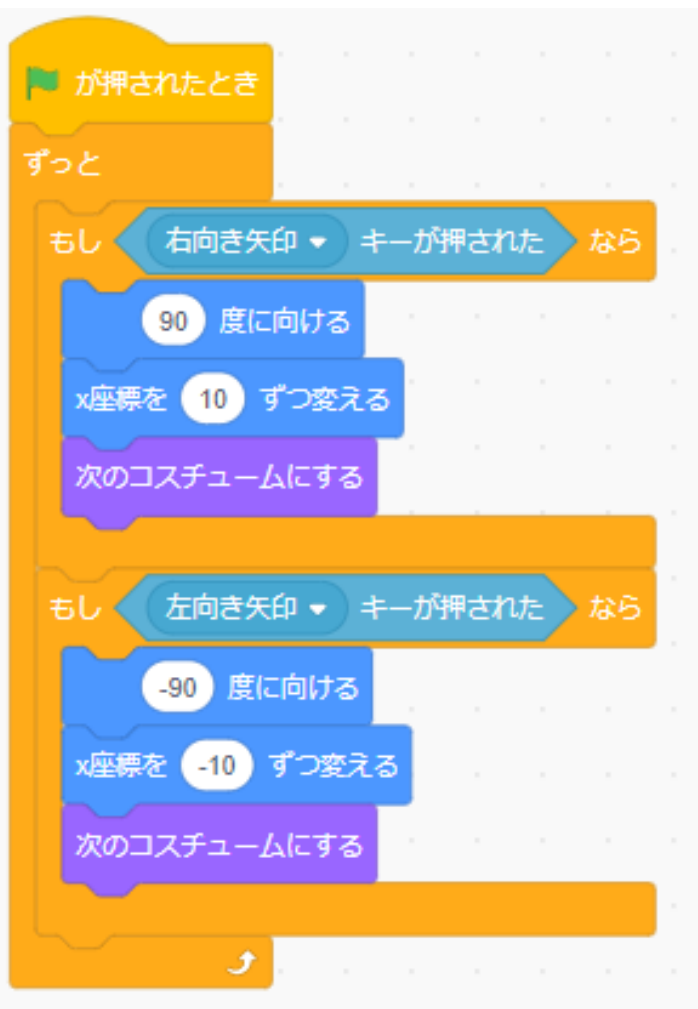
角度を「-90」にする



「左右のみ」を選択

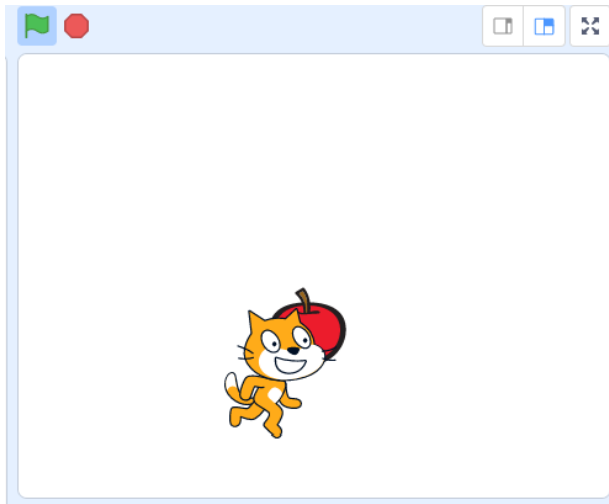
ネコを自由に動かす⑦

左右に動くプログラム



ジャンプするプログラム





ネコも動くようになったし
完成ね！

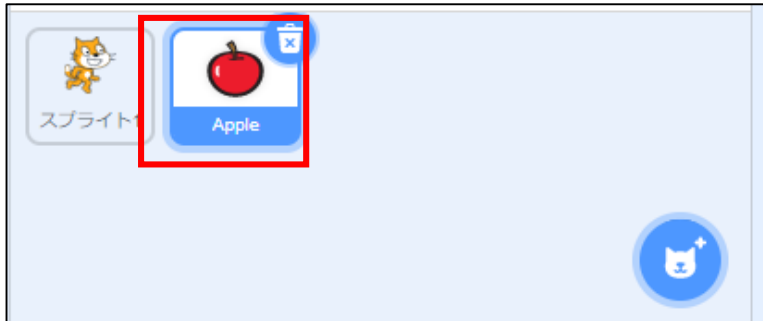
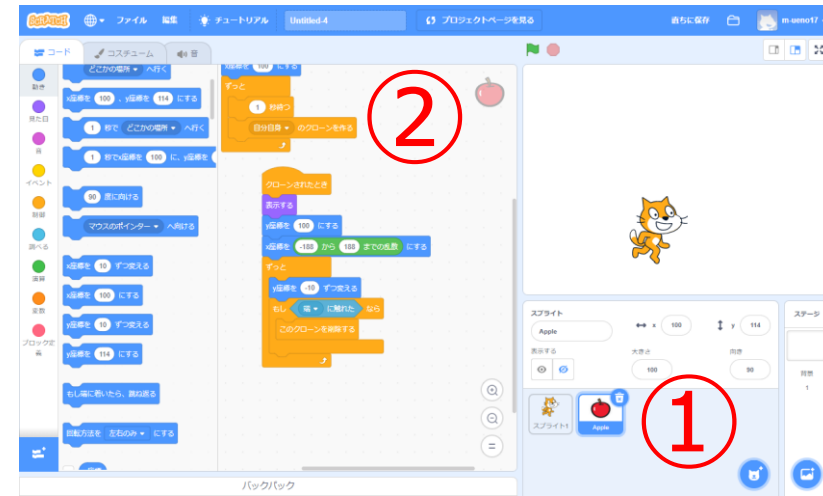


でも、ネコとリンゴは
すれ違いにや

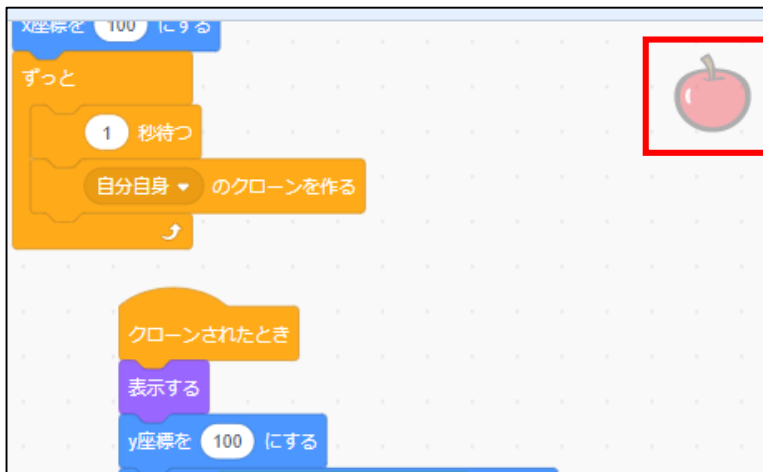
【Next Mission】

- ①ネコを自由に動かせるようにしよう！
- ②ネコが触ったら、リンゴが消えるようにしよう！
- ③「ネコが触ったら得点」・「地面に落ちたら減点」にしよう！

リンゴの編集画面に戻る



①画面右下で、ネコからリンゴに切り替える



②コード画面に、リンゴが表示されていることを確認する

ネコが触ったらリンゴが消える①

```
クローンされたとき
表示する
y座標を 100 にする
x座標を -188 から 188 までの乱数 にする
ずっと
  y座標を -10 ずつ変える
  もし 端 に触れた なら
    このクローンを削除する
```

「端に触れたら削除」のプログラムはできている…
同じ考え方で「ネコが触れたら削除」もできないだろうか？



「端」ではなく
「スプライト1」を
選択すればいい！

```
もし マウスのポインター に触れた なら
  このクローンを削除する
  ✓ マウスのポインター
  端
  スプライト1
```

ネコが触ったらリンゴが消える②

```
クローンされたとき  
表示する  
y座標を 100 にする  
x座標を -188 から 188 までの乱数 にする
```

ずっと

```
y座標を -10 ずつ変える
```

```
もし 端 ▼ に触れた なら
```

```
このクローンを削除する
```

```
もし スプライト1 ▼ に触れた なら
```

```
このクローンを削除する
```

➤

ネコが触れたら、リンゴが消えるようになったにや。
せっかくなので
点数も入るようにしたい！




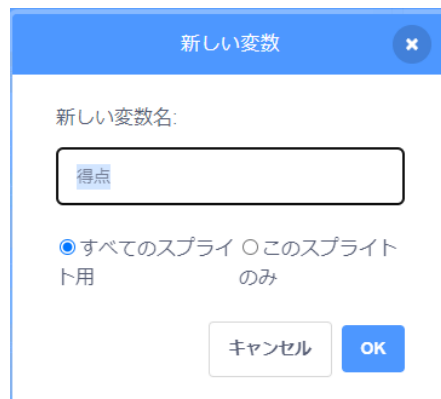
じゃあ、地面に落ちたら
減点だね！



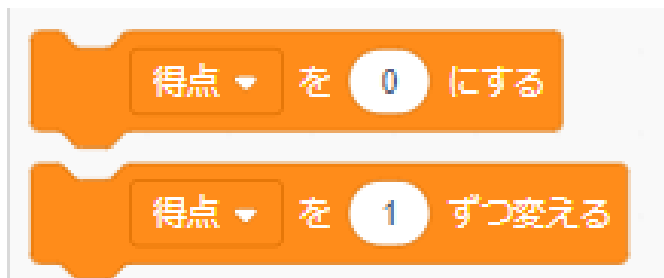
得点と減点を組み込む①



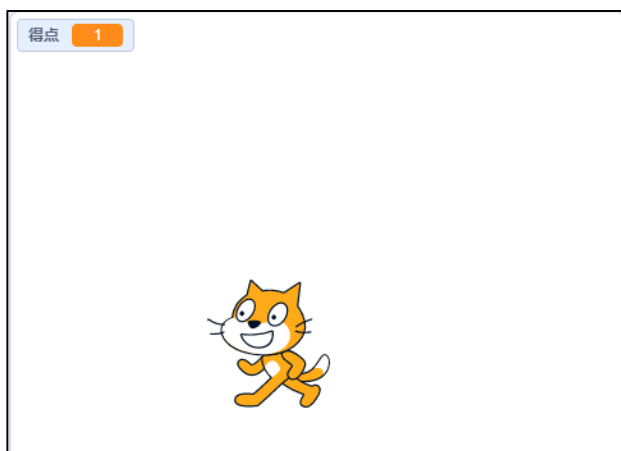
①  **変数** を選択し
 を
タップする



② 「得点」という
変数を作って
「OK」を
押します。



③ 「得点」のブロックが
できました！



④ ステージにも「得点」が
表示されました

得点と減点を組み込む②



《チャレンジ！》

- ・ 落ちたら「-1点」
- ・ ネコが触れたら「1点」



得点ブロックを、どこに入れたらよいでしょう？

得点と減点を組み込む③



• 落ちたら「-1点」

• ネコが触れたら「1点」

よりゲームらしくするために

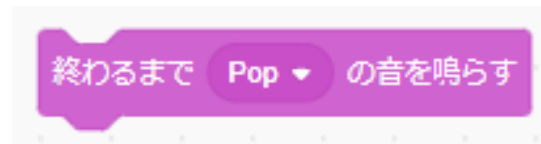


(1) 得点リセット

- ・左のプログラムを入れておくと旗を押すたびに、得点がリセットされます。

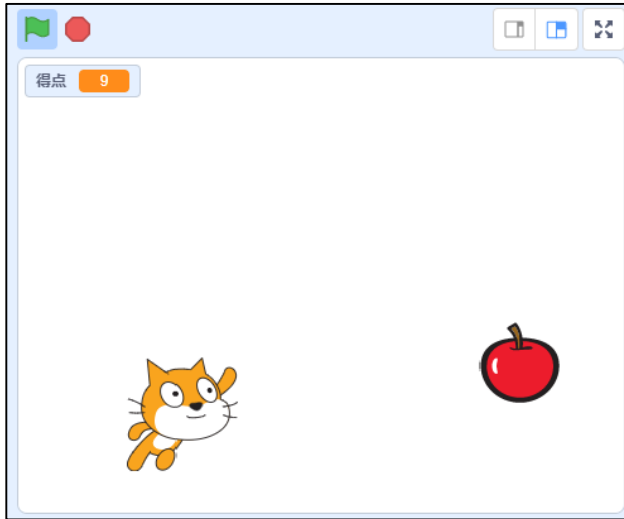


(2) 効果音



このブロックを入れておくとネコがリンゴに触れるたびに音が流れます。

作成したプログラミング



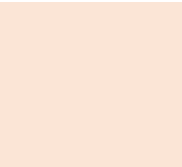
「ネコはリンゴが大好物！」

- ・リンゴが上からランダムに落ちてきます
- ・キーボードでネコを動かしてリンゴをキャッチ！
- ・キャッチできたら+1点
地面に落ちたら-1点

《協力》大分県次世代プログラマー発掘コンテスト
「Hello,World!」運営事務局

《目的》~~プログラミングを完成させること~~

- 必要な手順があることに気付く
- 記号の組合せをどのように改善していけば、より意図した活動に近づくのかを考える



プログラミング教材を
使ってみよう！

学習指導要領におけるプログラミング教育

小学校

- ・ 総則において、各教科等の特質に応じて、「**プログラミングを体験しながら**、コンピュータに意図した処理を行わせるために必要な論理的思考力を身に付けるための学習活動」を**計画的に実施することを新たに明記**
- ・ **算数、理科、総合的な学習の時間**において、プログラミングを行う学習場면을例示

↓
小中の連携が大事！

中学校

- ・ 技術・家庭科技術分野において、**プログラミングに関する内容を充実**（「計測・制御のプログラミング」に加え、「ネットワークを利用した双方向性のあるコンテンツのプログラミング」について学ぶ）

学習指導要領におけるプログラミング教育

小学校

- 算数、理科、総合的な学習の時間等で、活用できるプログラミング教材



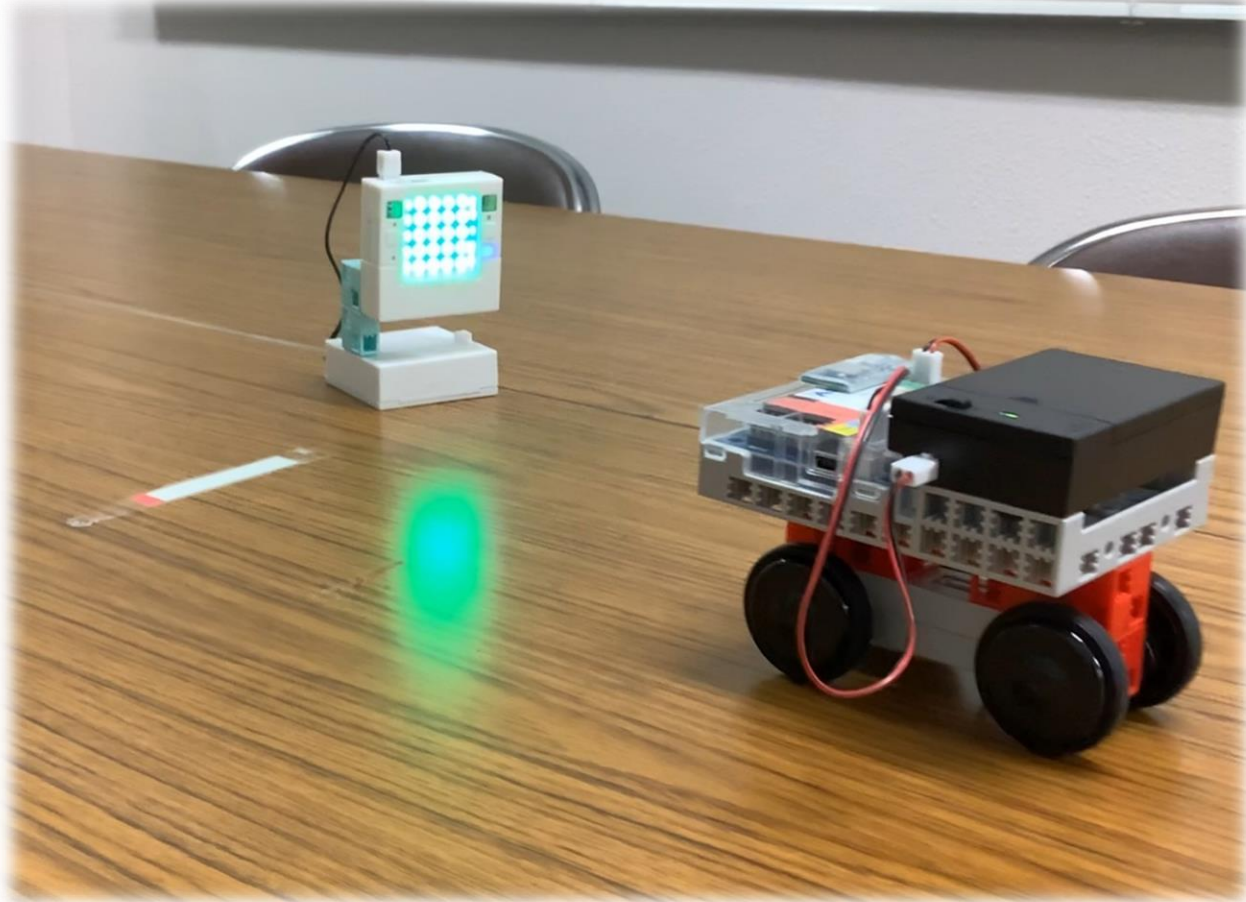
小中の連携が大事！

中学校

- 「計測・制御のプログラミング」
- 「ネットワークを利用した双方向性のあるコンテンツ」



中学校教材（2.0）で 信号機を作成



小学校教材（1.0）で
ロボットカーを作成

作成方法

(1) チームでチャレンジ

- ・小中合同のチームで作成します
それぞれの教材を使ってください

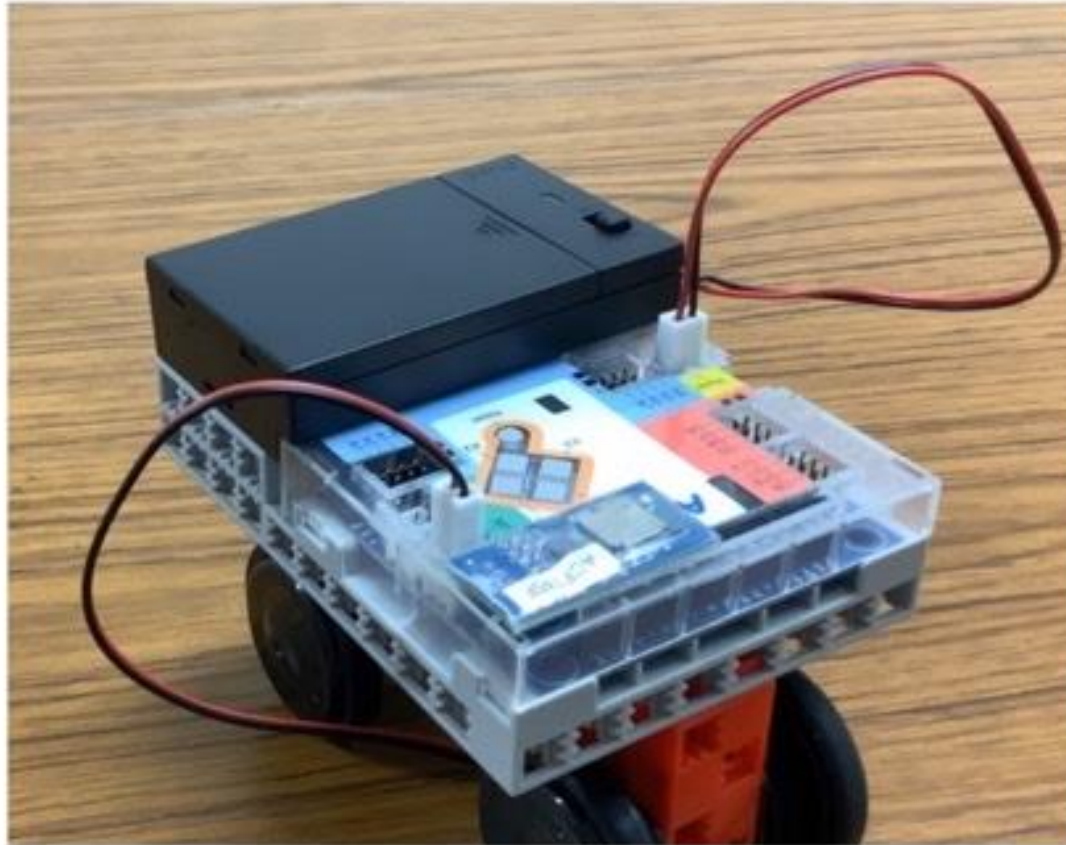
(2) 作り方はロイロ上で確認

- ・ロイロ上に組み立て方や
プログラムの内容をアップしています。



※ 「ロイロノートでログイン」を選択

学校 : oit000
I D : p番号
パス : pro0801



ロボットカー作り

ロボットカーの組み立て方①



【使う部品】

- スタディーノ
- 電池ボックス
- DCモーター
- タイヤ4個
- 赤ブロック5個



- (1) 赤ブロックとタイヤ2つを使います
- ①赤ブロックを組み立てる
 - ②赤ブロックにタイヤを取り付ける

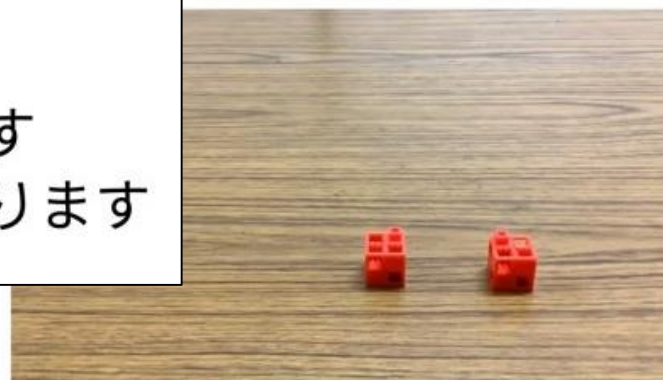
ロボットカーの組み立て方②



(2) DCモーター

②モーターにタイヤをつけます

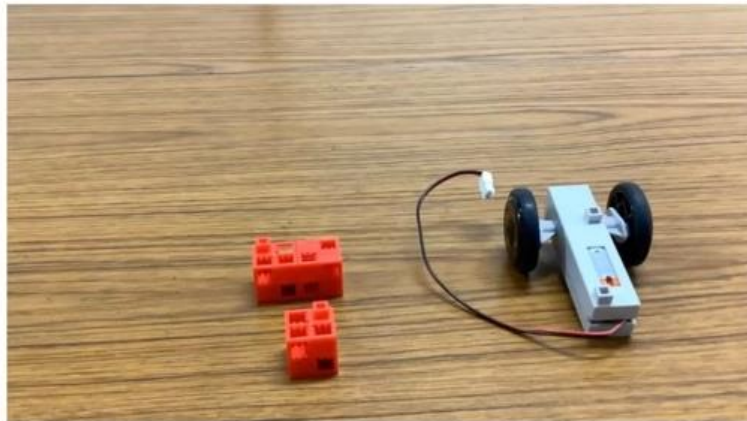
◆すでについている場合もあります



(3) 赤ブロック2個

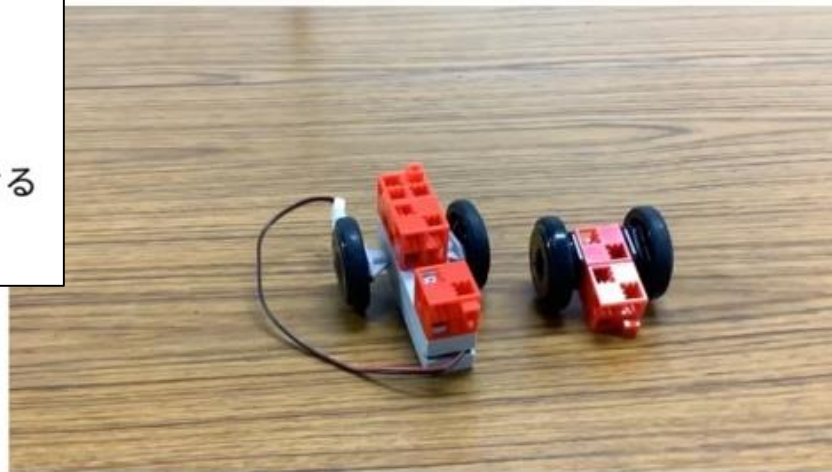
①2個のブロックを組み立てます

ロボットカーの組み立て方③



(4) DCモーターと赤ブロック

- ① (3) で組み立てた赤ブロックをタイヤ側につける
- ② 残り1個の赤ブロックを線側につける



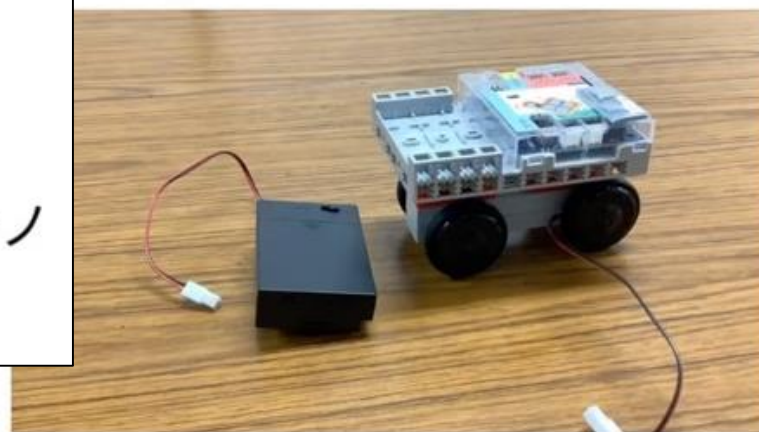
(5) タイヤ取り付け

- ① タイヤつきの赤ブロックをモーターに取り付けます

ロボットカーの組み立て方④

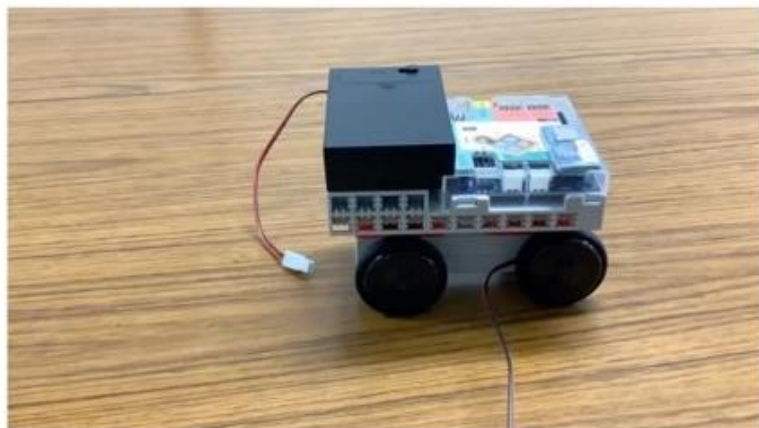


- (6) スタディーノに取り付ける
①タイヤつきモーターをスタディーノ
本体に取り付けます

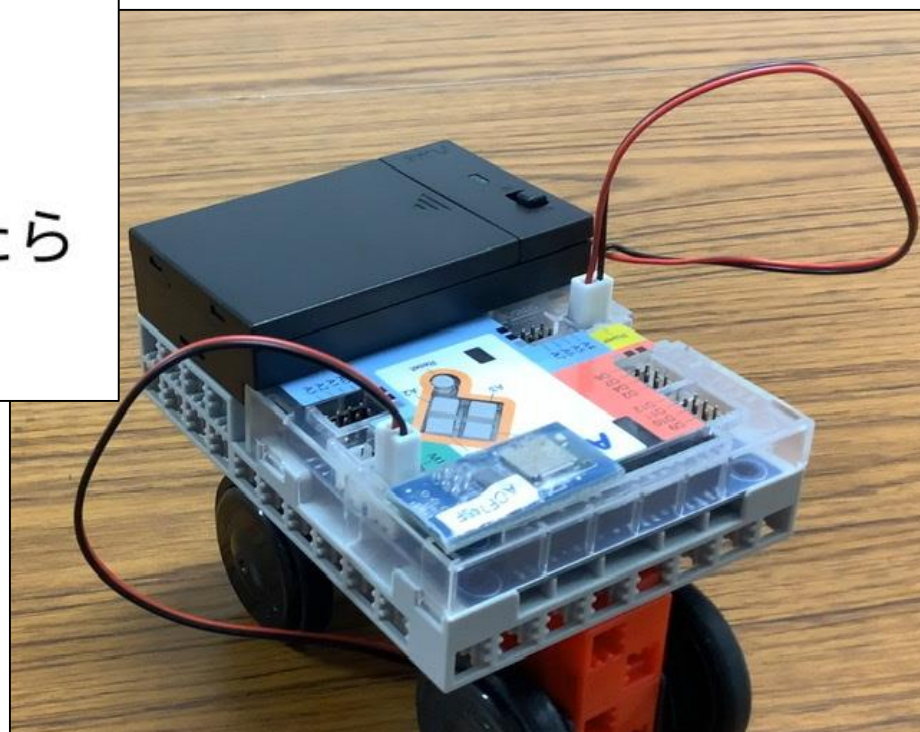


- (7) 電池ボックスを取り付ける
①向きに注意して、取り付けます

ロボットカーの組み立て方⑤



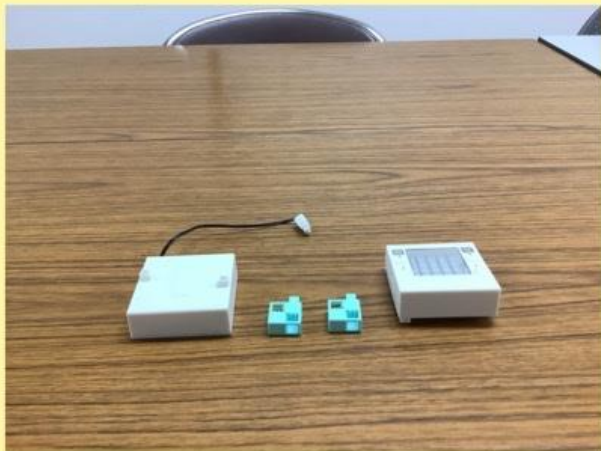
- (8) コードを取り付ける
① 向きに気をつけて取り付けたら
完成です！





信号機作り

信号機の組み立て方①



【使う部品】

- Studuino:bit ・ 電池ボックス
- ハーフブロック薄水 2 個



(1) ハーフブロックを組み立てる

信号機の組み立て方②

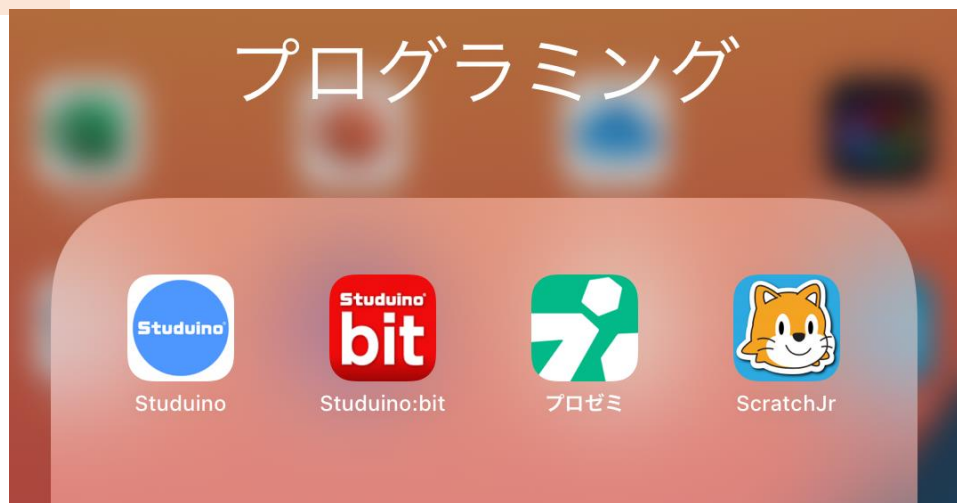


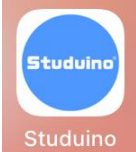
(2) 電池ボックスにブロックを固定する



(3) Studuino:bitを固定して完成

iPadとの接続方法① (小 = 1.0)



①  をタップします。


どちらのモードで作成するか選択してください

キャラクターモード

電気実験モード

画面内でキャラクターを動かすプログラムや、アーテックロボを動かすプログラムを作成できます。

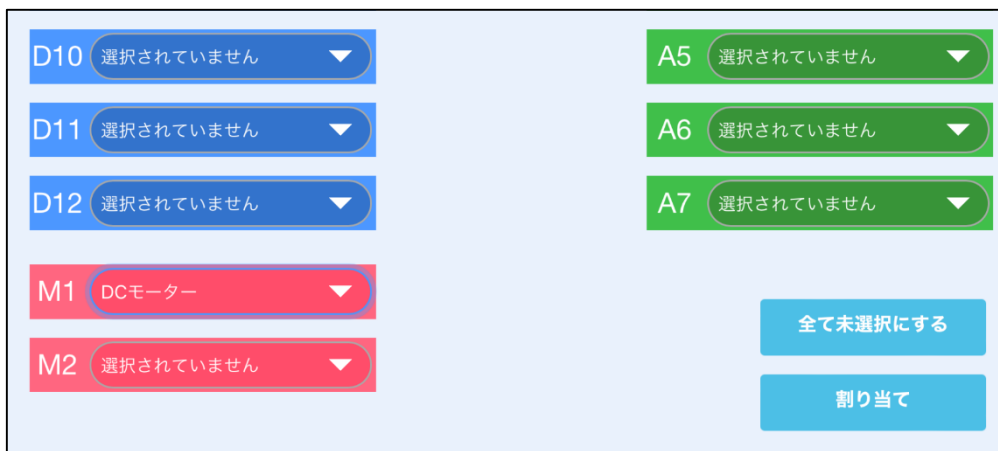
電気の流れをセンサーによって制御するプログラムを作成できます。

②  を
タップします。

iPadとの接続方法②（小=1.0）

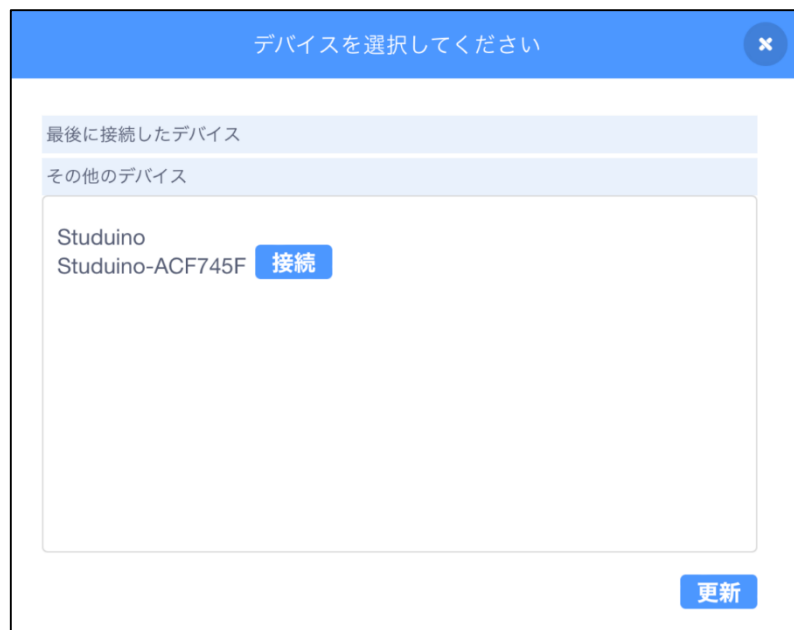


③ 「編集」→「入出力設定」を選択します。

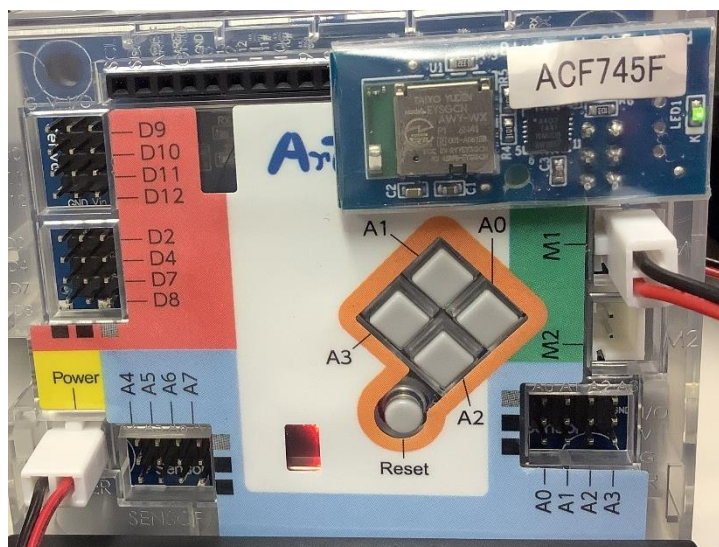


④ 「M 1」の場所に「DCモーター」を選択し、「割り当て」をタップします。

iPadとの接続方法③ (小 = 1.0)



⑤ 「編集」→「接続」をタップし自分のデバイスと接続します。



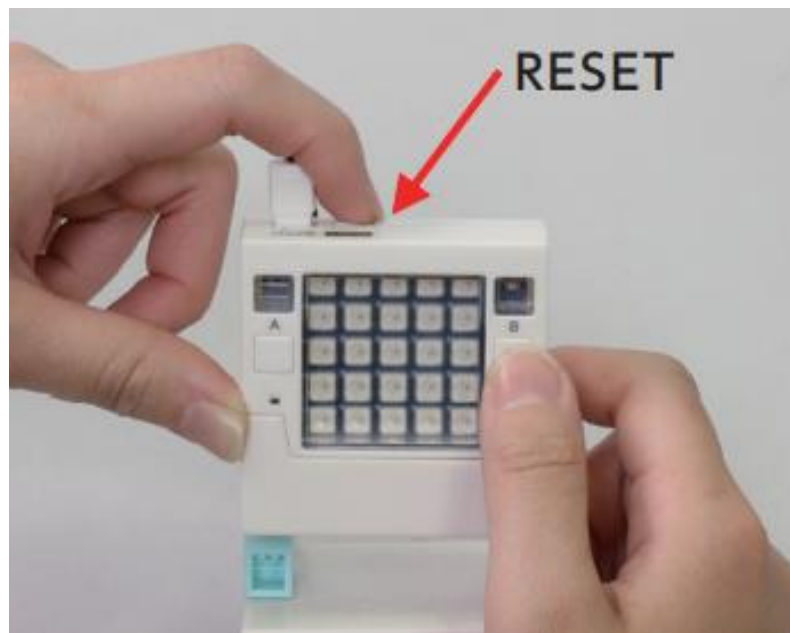
本体に貼られているシールと同じ番号を選びます

iPadとの接続方法① (中 = 2.0)

①「編集」から接続を選択します。



②表示されるメッセージに従い、メインユニット (Studuino:bit) の B ボタンを押しながらリセットボタンを押してください。



iPadとの接続方法② (中 = 2.0)

- ③メインユニット (Studuino:bit) の LED に表示された点灯パターンと同じデバイスを選択してください。



プログラム（例）



ロボットカー



信号機

小学校でプログラミング教育を 必修化した理由

便利な機械が「魔法の箱」ではなく、**プログラミングを通じて人間の意図した処理を行わせることができるもの**であり、人間の叡智が生み出したものであることを理解できるようにすることは、時代の要請として受け止めていく必要がある。

小学校段階におけるプログラミング教育の在り方について
(議論のまとめ) 平成28年6月16日