

# [応用編(A)]

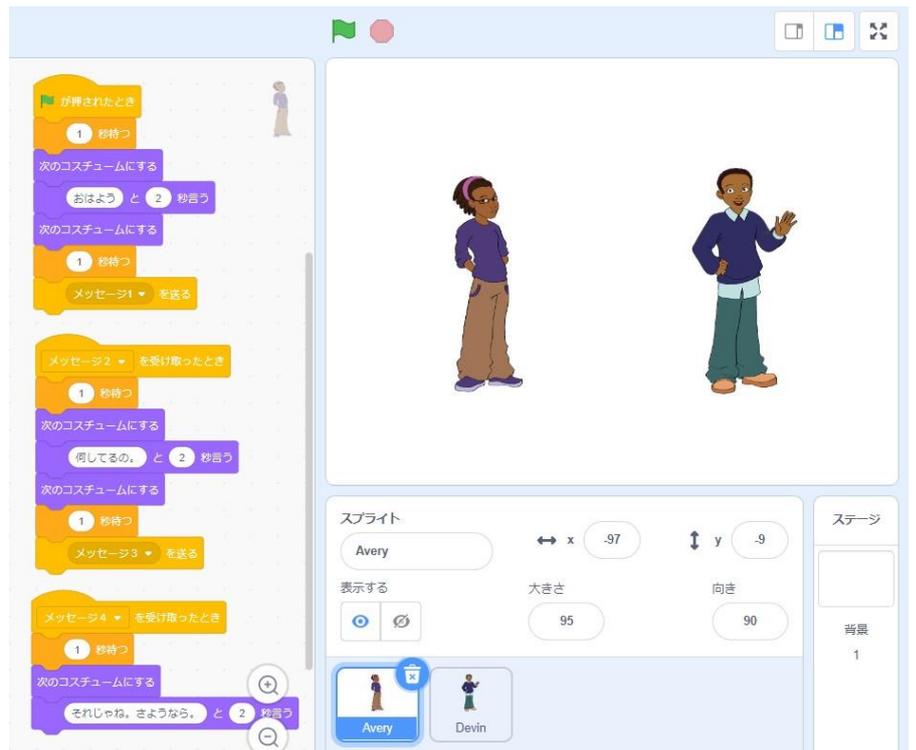
# 1 お話をする

## 1 お話をする

男の子と女の子が  
話をしています。

右のページは女の子が  
話しているプログラム  
です。

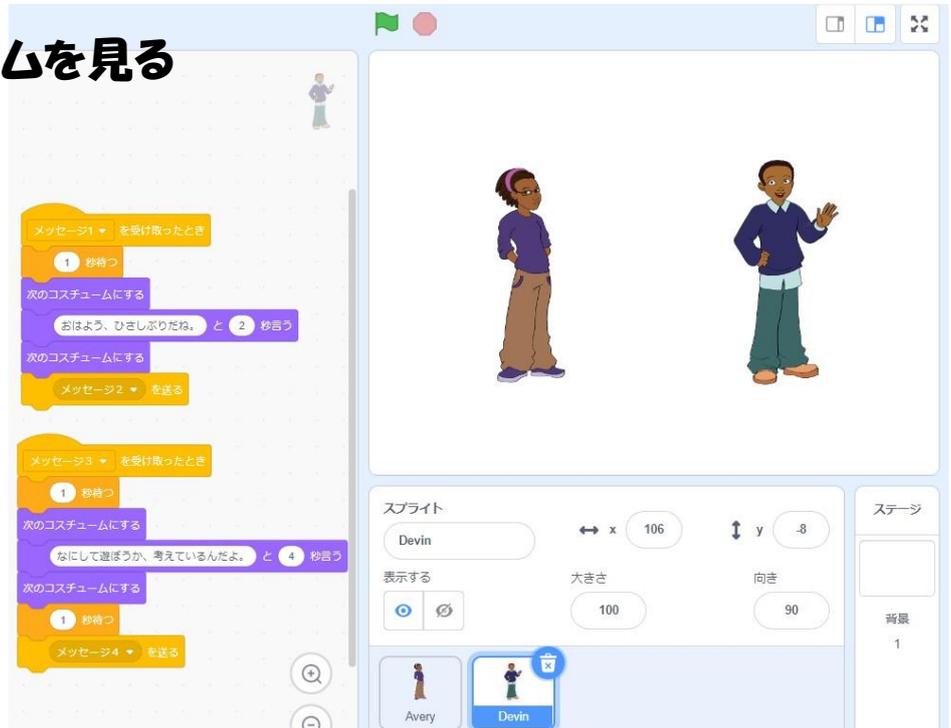
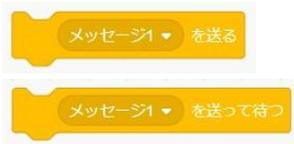
次のページは男の子が  
話しているプログラム  
です。



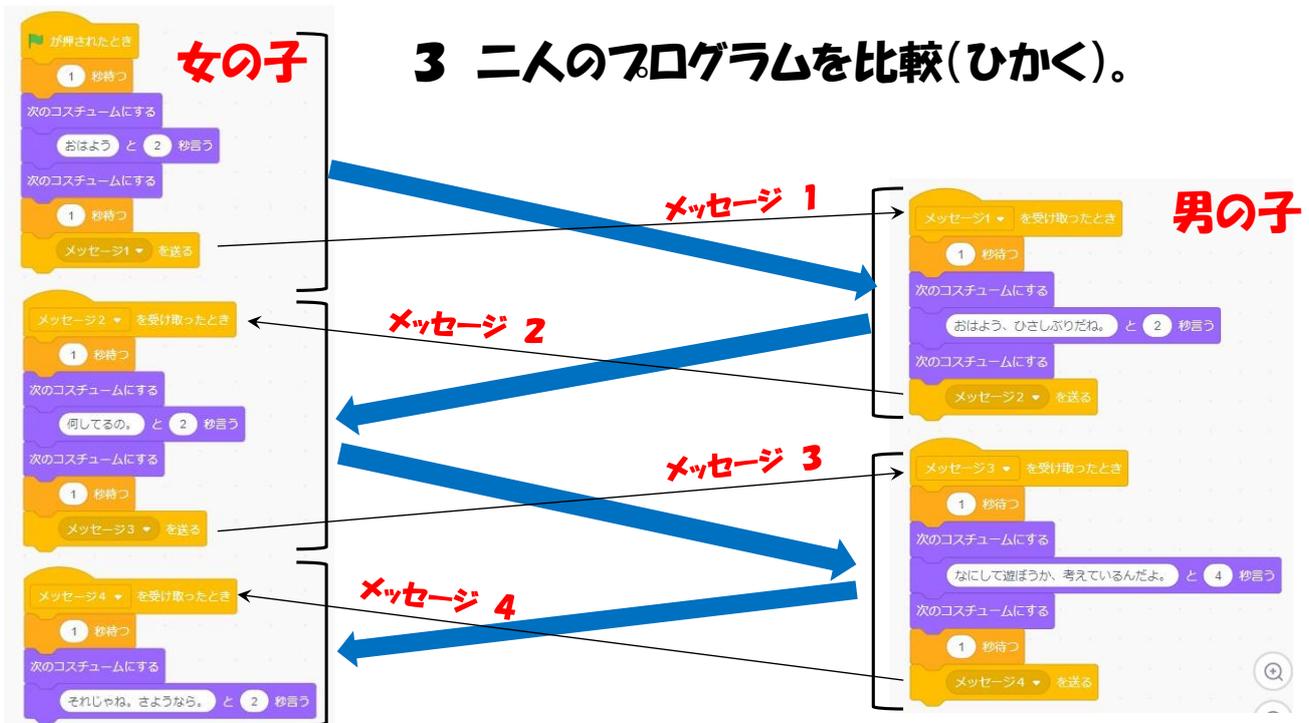
## 2 お話のプログラムを見る

次のページに、  
二人のプログラムを  
並べて書いています。

このお話のプログラム  
では、下の二つの命令  
で話しているように、**タ  
イミング**を合わせてい  
ます。



## 3 二人のプログラムを比較(ひかく)。



(おわり)

## 2 もし石にふれたら

いろいろな作品を作る場合、「・・・にふれたら○○○」  
という動作が必要になる事があります。

この時に、どのようなプログラムに  
なるのか考えてみましょう。

1 この「もし石にふれたら」のプログラムは、ネコが歩いていて、もし石に触れたら「いた！」と言うプログラムです。

「もし～でなければ～」の命令を使ってみましょう。



ねこが(Rocks)に触れたら、「いた!」と言います。

そうでなければ、ねこは歩き続けます。

「～に触れた」を調べるためには、色を使った命令を使うことがあります。

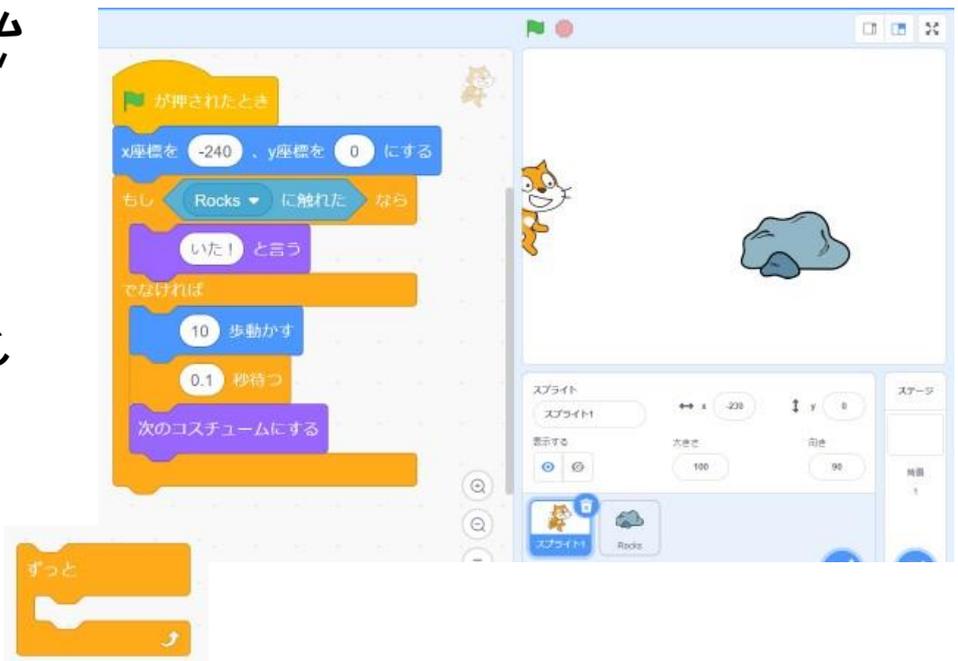
2 それでは、全体のプログラムを組み立ててみましょう。

それでは、全体のプログラムを組み立て、緑の旗をクリックしてみましょう。

うまく動きませんね！  
なぜでしょう。

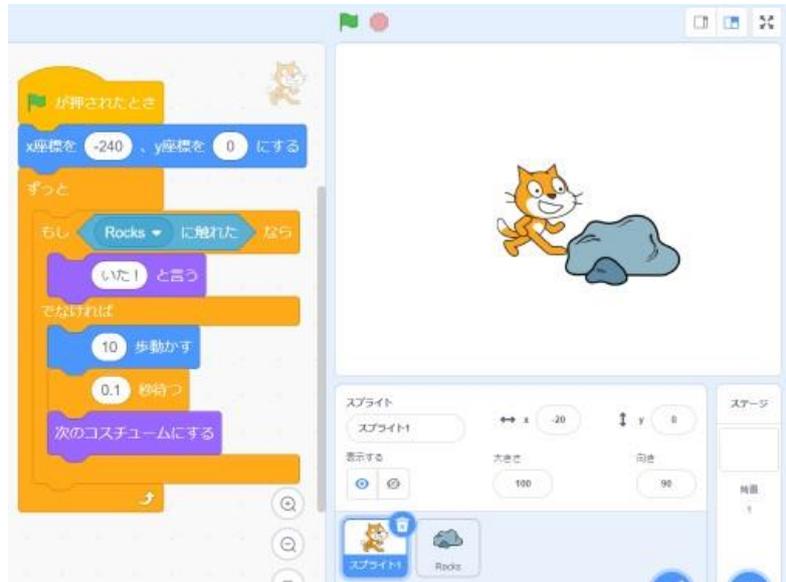
それは、右のプログラムが、  
上から下へ一度だけ動作し  
止まってしまったからです。

それでは、「もし～でなければ～」命令を「ずっと」命令で囲んでみましょう。



### 3 プログラムの完成。

これで、ねこが  
(Rocks)に触れたら、  
「いた！」というプロ  
グラムが完成しました。



(おわり)

## **3 ネコのジャンプ。**

**ゲームでは、主人公を「ジャンプ」  
動作をさせることがあります。**

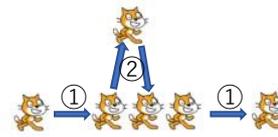
**そこで今回は、ネコを「ジャンプ」  
させてみましょう。**

# 1 スペースキーを押し、ネコをジャンプさせよう。



① 先ずネコが左右に連続して動くプログラムです。このプログラムは、もう何度も勉強しましたね。

② このプログラムは、「スペースキーが押されたとき」その次の命令が実行されます。その命令とは、「Y座標を60ずつ変える」を実行しネコは上の方向へジャンプします。その後「0.1秒待つ」た後に、「Y座標を-60ずつ変える」を実行し、下の方向へ着地（ちゃくち）します。この「スペースキーが押されたとき」の命令はいつ実行されるか分からないので、「ずっと」命令でかこんで、いつ押されてもよいように準備しています。



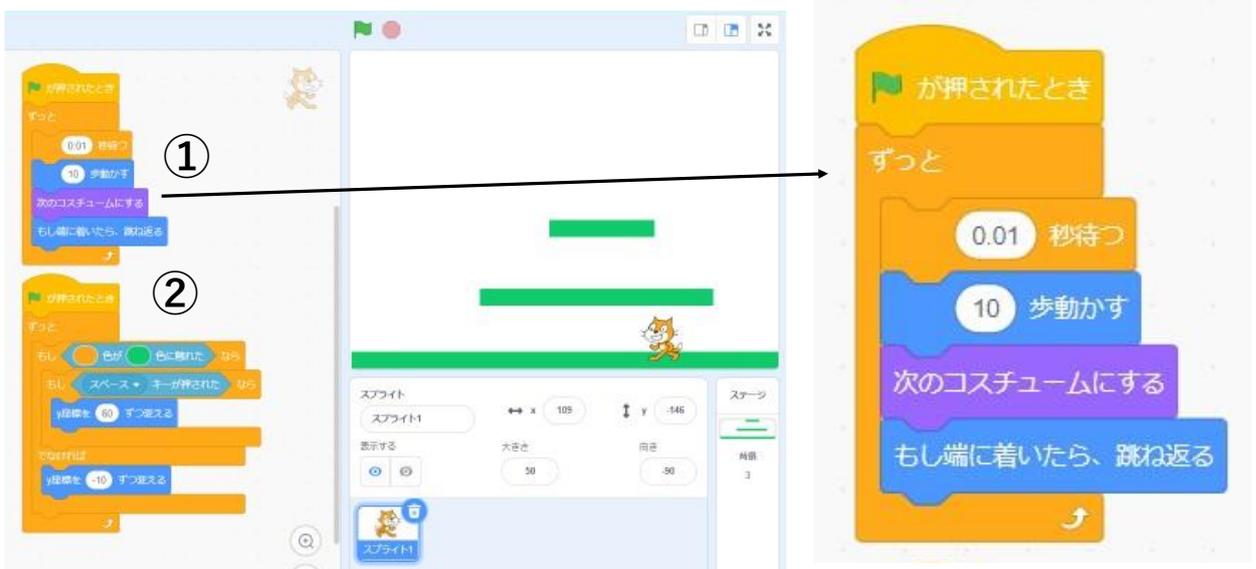
③ ネコがジャンプするプログラムは、右にある例などいろいろなプログラムが考えられます。



# 2 ネコに緑の階段を上り下りさせよう。

## ①のプログラムの説明

先ず、①のネコが左右に歩き続けるプログラムは同じですね。



## ②のプログラムの説明

Scratch program logic for a cat jumping:

- When green flag clicked
- Forever loop:
  - If color touches color:
    - Change y-coordinate by 60
  - If space key pressed:
    - Change y-coordinate by -10

Diagram labels:

- (ロ) Cat on the lower platform.
- (イ) Cat jumping from the lower platform to the higher platform.

(おわり)

# 4 シューティングゲーム

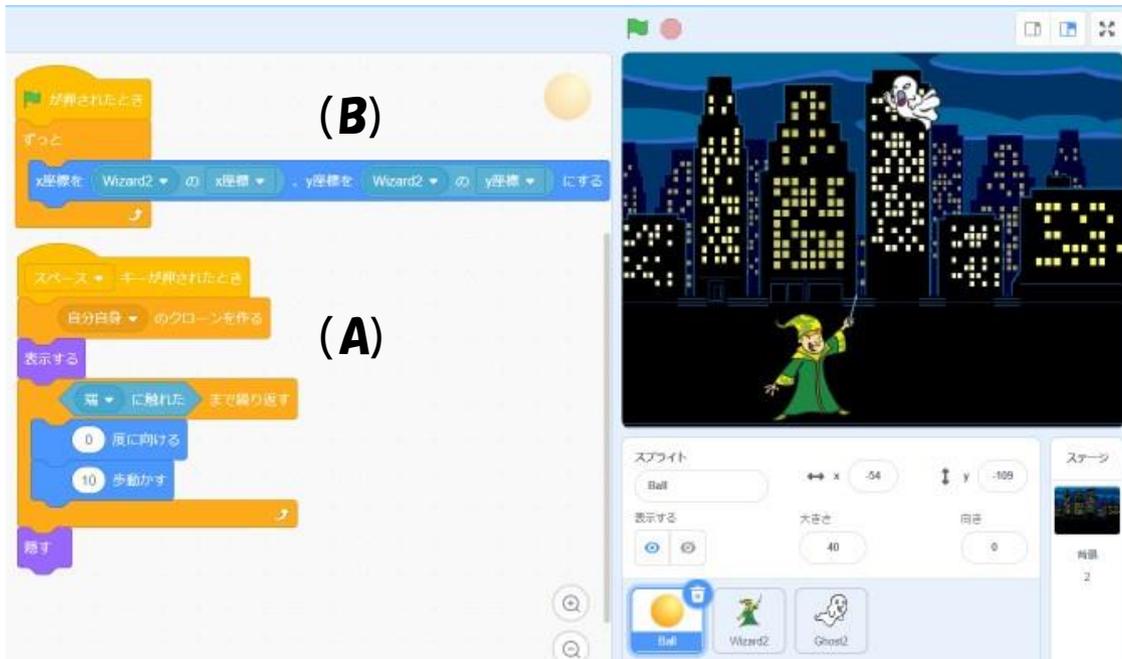
## シューティングゲーム

ゲームにはいろいろな種類がありますが、その中でもシューティングゲームは一度は経験したことがあることでしょう。今回は、このゲームの基礎になるプログラムについて説明しましょう。

シューティングゲームには、次の二つのプログラムがポイントになります。

- 弾(たま)を発射(はっしゃ)するプログラム
- 弾が当たったことを検出するプログラム

# 1 弾(たま)のプログラム



(A) ここでは、スペースキーをクリックすると、次々と弾(たま)が飛び出すプログラムを説明しましょう。

(B) それとともに、魔法使いが左右に動くと、それと合わせて弾の出る所も左右に動くプログラムを説明しましょう。

## (A) 弾(たま)を発射(はっしゃ)するプログラム

- (1) スペースボタンがおされた時に、このプログラムが動き始めます。
- (2) このプログラムは弾のプログラムですから、自分自身とは弾のことです。この弾のクローン(同じもの)をスペースボタンがおされたら作ることにします。
- (3) (2)のクローン(同じもの)でつくられ弾に見えるように表示します。
- (4) 弾が画面の端(はじ)にふれるまで、(5)(6)の命令をくり返します。
- (5) 0度とは、Scratchでは上の方向を指します。
- (6) いつもの「10歩」歩く命令です。(4)(5)(6)の3つの命令で、弾がうえの方向へはじにふれるまで発射(はっしゃ)されます。
- (7) 発射された弾が、はじに着くと弾が消えます。

### 弾(たま)のプログラム



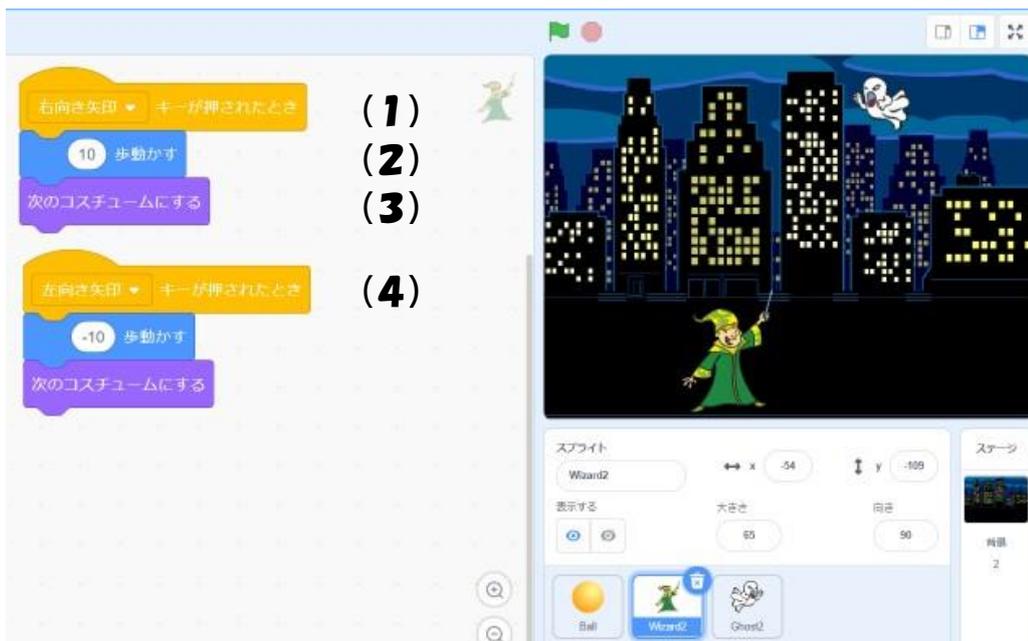
## (B) 弾(たま)を発射(はっしゃ)する所を変えるプログラム



- (1) 緑の旗(はた)が押された時に、このプログラムが動き始めます。
- (2) (3)の命令がずっとくり返されます。つまり(3)の命令が常に動作していることになります。
- (3) この命令は、弾(たま)の位置(座標)と、まほう使いの位置(座標)を同じにする命令です。

弾のX座標 = まほう使いのX座標 → 弾の座標 → (X座標(まほう使いのX座標)、Y座標(まほう使いのY座標))  
 弾のY座標 = まほう使いのY座標  
Wizard2 Wizard2

## 2 魔法使いのプログラム

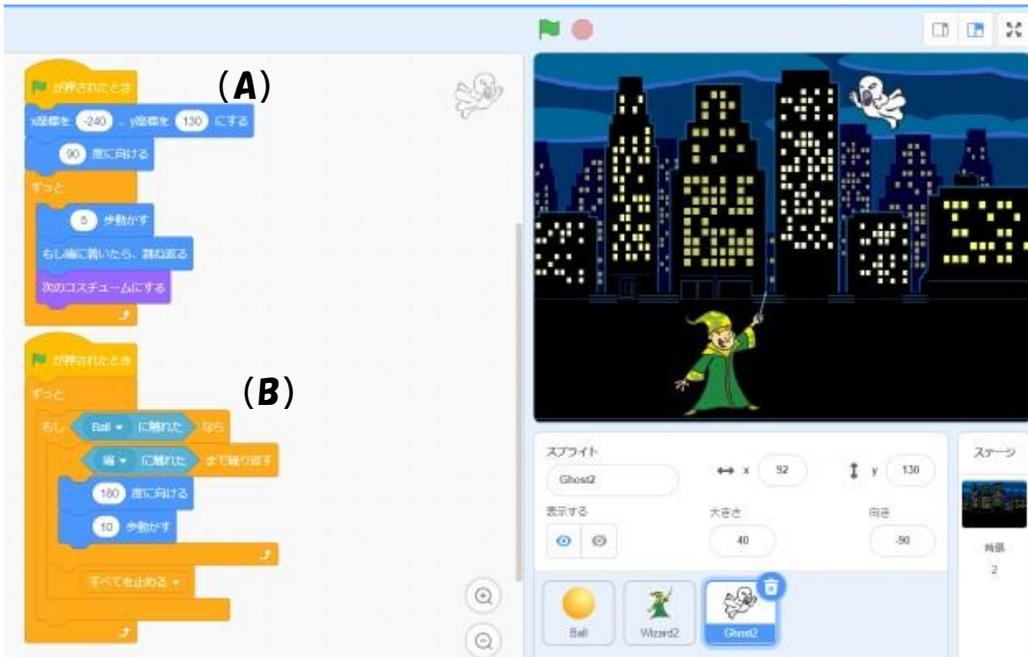


ここでは、魔法使いが左右に動くプログラムを説明しましょう。

- (1) 「右向き矢印キーが押されたとき」
- (2) 「10歩動かす」命令で、10歩右の方向へ歩きます。
- (3) 「次のコスチュームにする」命令で魔法使いの形が少し変わります。
- (4) 「左向き矢印キーが押されたとき」も同様ですが、左向きですから、「-10歩」となります。

### 3 おぼけのプログラム

ここでは、おぼけが動くプログラムの説明です。



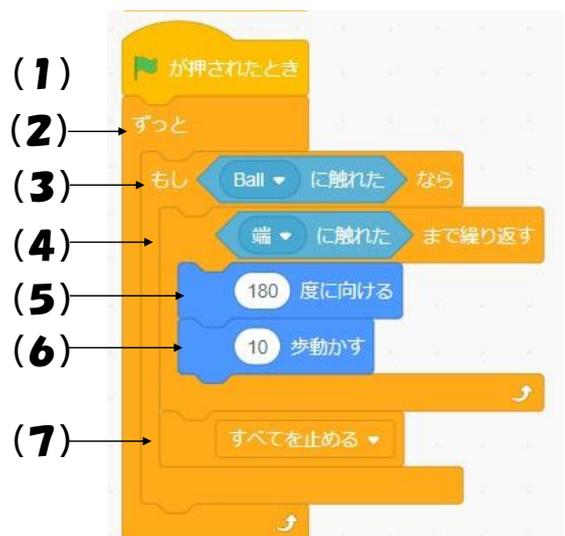
(A)は、おぼけが左  
右に動くプログラム  
です。

これはすでに何度  
も作ったプログラム  
ですね。  
「90度に向ける」と  
は、右の方向を向く  
とのこと。

### (B) 弾が当たったことを検出するプログラム

- (1) 緑のはたがクリックされた時に、このプログラムが動き始めます。
- (2) このプログラムは弾が標的(ひょうてき)に当たった時の標的のプログラムですから、いつ起きるかわかりません。そのため「ずっと」の命令で全体をかこんで、常にかんじています。
- (3) もし、この標的がBall(ボール・弾)に当たったら、(4)～(7)の命令を実行します。
- (4) まず、標的ははじにふれるまで、(5)～(6)の命令を実行します。
- (5) 180度とは、Scratchでは下の方向を指します。
- (6) いつもの「10歩」歩く命令です。(4)(5)(6)の3つの命令で、標的が弾に当たった時に、標的ははじにふれるまで下の方向に動いていきます。
- (7) 標的が下のはじに着いた後に、すべてを止めます。

弾に当たった時の標的(ひょうてき)のプログラム



(おわり)

## 5 迷路(めいろ)を進め

**「迷路(めいろ)を進め」**

皆さんは、迷路(めいろ)の中をボールがどこにも触(さ)わずに最後のゴールまでたどり着く、というゲームを試したことがあることでしょうか。

ここでは、非常にシンプルな「迷路」を設定(せてい)し、基本的なプログラムとその動きについて説明します。

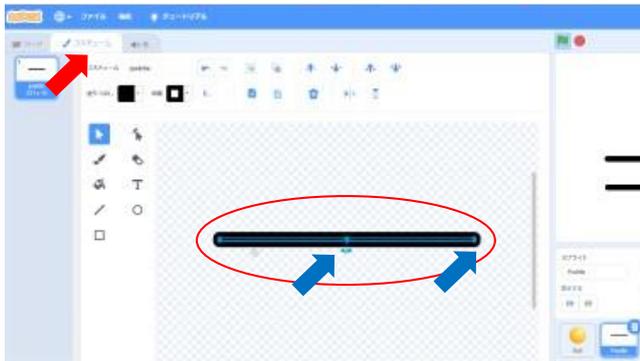
皆さんはこれを学んで、いろいろな種類の更に複雑な「迷路を進め」のプログラムを作ってみてください。

## (1) 迷路を作る

自分の好きな迷路を作ってみましょう。

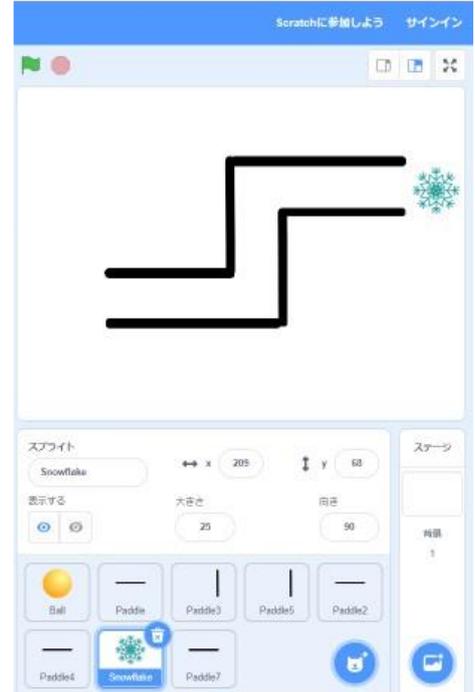
迷路の色は、黒などを使いましょう。うすい色を使うと、パドルをコピーする時に微妙(びみょう)に色がちがひ、球(たま)が迷路の壁(かべ)を認識(にんしき)しないことがあります。

- ① スクラッチに登録(とうろく)してある、Paddle(パドル)の形をダウンロードし、このパドルを6本コピーし迷路を作ります。



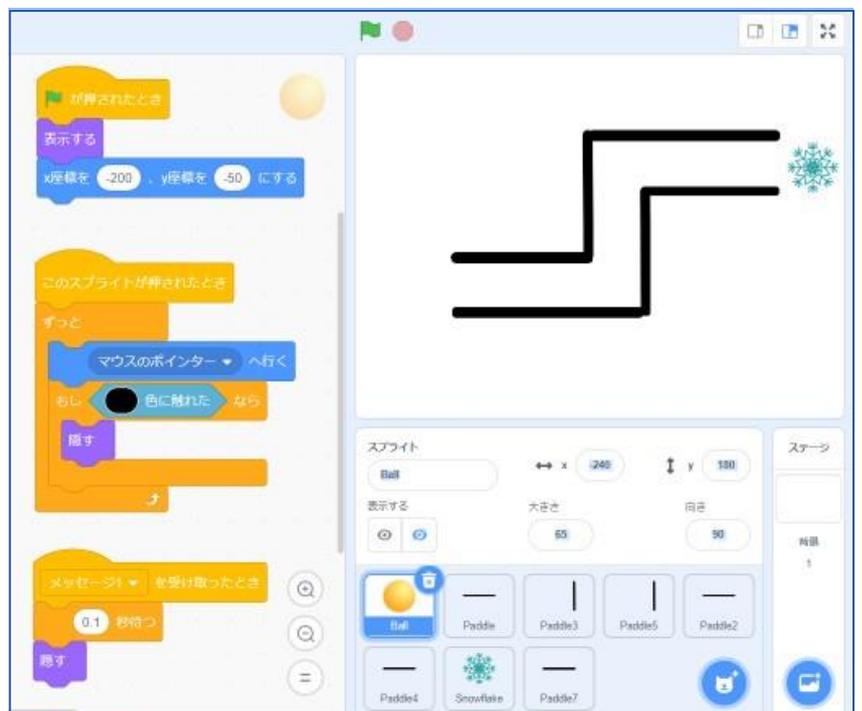
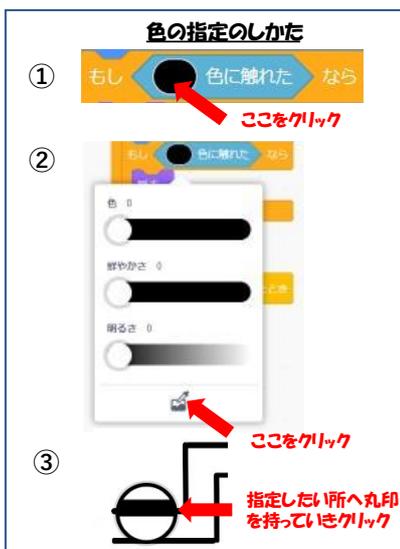
- ② ダウンロードした後に、コスチューム(赤矢印)をクリックすると上図となります。この図の赤い円のなかの青矢印を操作(そうさ)すると、パドルの長さや傾(かたむき)を変えることができます。

- ③ 自分の好きな迷路を作ってみましょう。



## (2) ボールのプログラムを作る

スクラッチに登録(とうろく)してあるボールをダウンロードし、そのボールを動かすプログラムを作ってみましょう。



## ボールのプログラムの説明

**が押されたとき** → **旗がクリックされた時、**  
**表示する** → **ボールが表示(ひょうじ)されます。**  
**x座標を -200、y座標を -50 にする** → **その表示されるボールの位置は**  
 $X座標(ざひょう) = -200$   
 $Y座標(ざひょう) = -50$

**このスプライトが押されたとき** → **このスプライトとは、ボールのことです。従ってボールがクリックされた時、次の命令が実行されます。**  
**ずっと** → **「ずっと」命令では含まれた3つの命令をずっとくり返します。**  
**マウスのポインター へ行く** → **マウスのポインターが動くとおりに、ボールが動きなさいという命令です。**  
**もし 色に触れた なら** → **もし、ボールが黒色にさわったなら、[隠す(かくす)]の命令を実行(じっこう)しなさいという命令です。**  
**隠す** → **ボールを隠(かく)すとは、ボールが消えなさいという命令です。**

**このひとまとまりのプログラムで、ボールがマウスのポインターの通りに動き、黒色のかべにさわったら、ボールが消えることになります。**

**メッセージ1 を受け取ったとき** → **「メッセージ1」を受け取った時 (「メッセージ1」は、後で出てきます。)**  
**0.1 秒待つ** → **0.1秒の短い間、そのまま待ちます。**  
**隠す** → **その後、ボールが消えなさいという命令です。**

## (3) ゴールのプログラムを作る

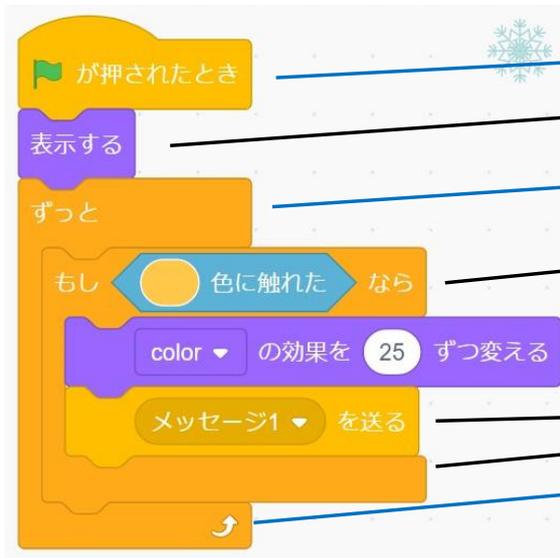
ゴールの印をダウンロードし、このゴールの印(ここでは雪の形)へプログラムを書きこみましょう。

The screenshot shows the Scratch interface with a snowflake sprite selected. The code blocks for the snowflake are:

- が押されたとき** (When clicked)
- 表示する** (Show)
- ずっと** (Forever loop) containing:
  - もし 色に触れた なら** (If color touched):
    - color の効果を 25 ずつ変える** (Change color effect by 25)
    - メッセージ1 を送る** (Send message 1)

The stage shows a yellow ball and a black goal shape. The snowflake is positioned at the top right of the goal. The sprite panel shows the snowflake is selected.

## ゴールのプログラムの説明



がクリックされた時、

ゴールの絵が表示されます。

「ずっと」命令ではさまれた3つの命令を  
ずっとくり返します。

もし、ゴールの絵がボールの色にさわったなら、間にはさま  
れた2つの命令を実行(じっこう)なさいという命令です。

色が次々に変わいなさいという命令です。

「メッセージ」という信号を送ります。  
(2)のボールのプログラムとつながります。

(おわり)

## せいげんじかん　せってい 6 制限時間を設定する

ゲームを作る場合、制限（せいげん）時間を設定（せってい）し、その時間内のポイントを競う（きそう）ゲームが多くあります。

その制限時間を設定する  
プログラムを学びましょう。

**1** スクリプトの中の、[変数]をクリックします。



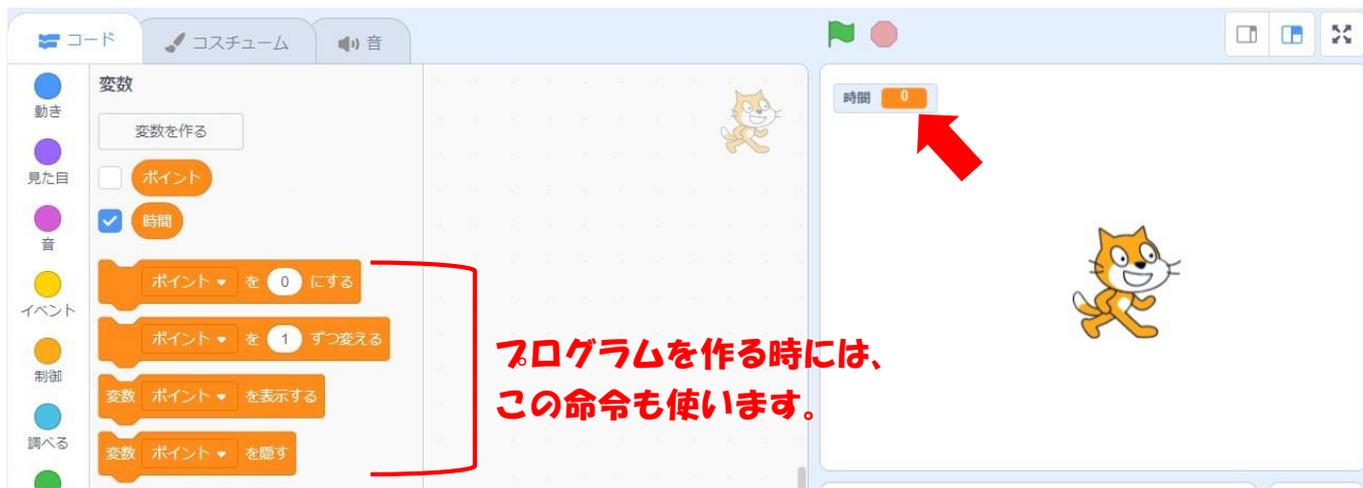
**2** 次に、[変数を作る]をクリックします。



**3** すると、下のように「新しい変数」という窓（まど）があらわれます。その「変数名」の所へ[時間]と入力し、[OK]をクリックします。



- 4 下のような画面へ変わります。  
この画面の右の赤矢印のところに、時間が表示されます。



- 5 それでは、制限時間を表示するためのプログラムはどのようなになっているか見てみましょう。

- ① 緑のはたがクリックされた時、同時にこのプログラムが動き始め、つまり時間を計算し始めます。
- ② [時間を10にする]とは、時間を10秒に設定(せってい)したことになります。
- ③ 時間が0になるまで、③の命令ではさまれた、④と⑤の命令を何度も実行します。
- ④ この命令で1秒間待ちます。
- ⑤ 時間の設定を1秒減らします。
- ⑥ ③の命令で、時間が0秒になった後、時間のカウントも含め、全てを止めます。



(おわり)

# 7 ポイントを数える

**ゲームを作る場合、制限（せいげん）時間を  
設定（せってい）し、その時間内のポイントを  
競う（きそう）ゲームが多くあります。**

**そのポイントを数える  
プログラムを学びましょう。**

**1** スクリプトの中の、[変数]をクリックします。



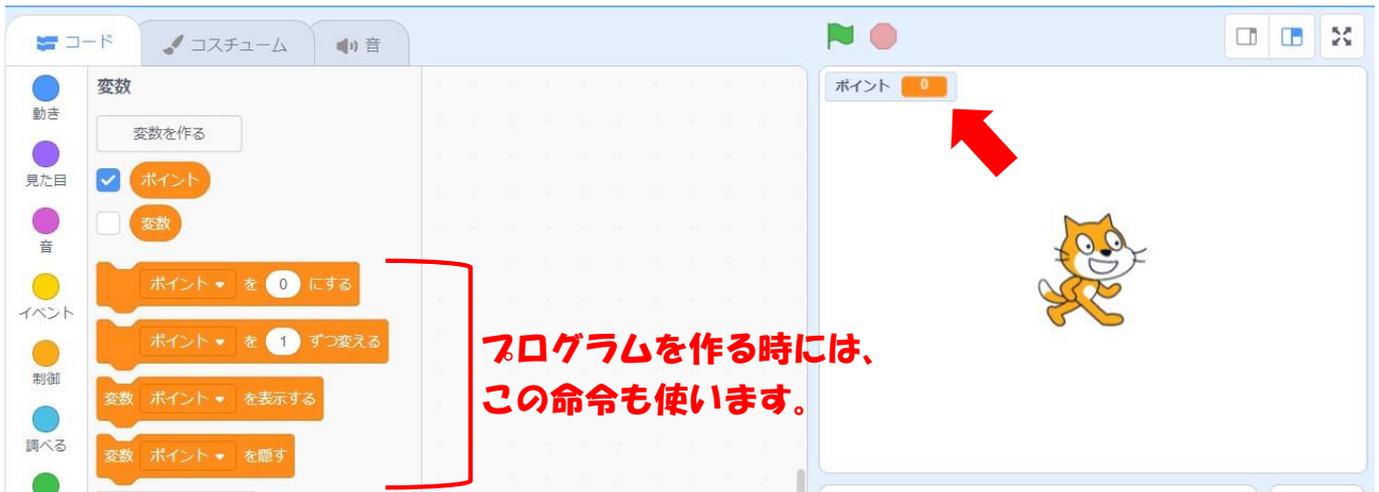
**2** 次に、[変数を作る]をクリックします。



**3** すると、下のよう「新しい変数」という窓（まど）があらわれます。その「変数名」の所へ[ポイント]と入力し。[OK]をクリックします。



- 4 下のような画面へ変わります。  
この画面の右上の赤矢印のところに、ポイントが表示されます。



## 5 ポイントを数えるプログラム

さてポイントと言っても、ゲームによってその定義(ていぎ)はいろいろとあるものです。

例えば、

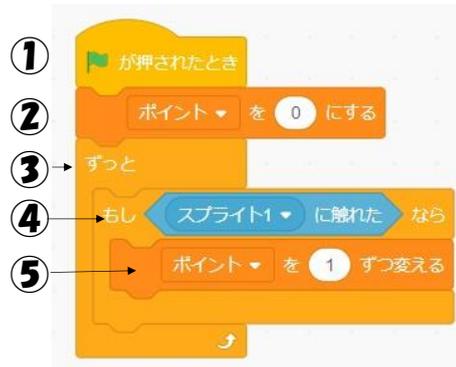
- (1) 標的(ひょうてき)にさわった数。
- (2) 相手を倒す(たおす)ために、クリックした数。

などが考えられます。

それでは、各々について、プログラムがどうなっているのか見てみましょう。

## (1) 標的(ひょうてき)にさわった数をチェックする。

- ① いつもの最初の命令です。
- ② ポイントを最初に「0」に設定(せってい)します。
- ③ 「ずっと」命令で、④と⑤の命令をはさんでいます。つまり、**ずっと常に④と⑤の命令を実行し続けています。**では、④と⑤はどのような命令なのでしょう？
- ④⑤ 「もし」登場人物が[スプライト1]にふれたなら⑤の[ポイント $\downarrow$ を1ずつ変える(増やすとの意味)]という命令です。この④と⑤の命令が常に実行し続けて、ポイントが増えることをチェックしているのです。



\* [スプライト1]という名前は、ゲームによりいろいろと変わります。

## (2) 相手を倒す(たおす)ために、クリックした数。

- ① このような場合、(変数)-(変数を作る)の変数として、[クリックした回数]を作る。
- ② プログラムとしては、[クリックした回数を1ずつ変える。]を使用する。



(おわり)

## 8 落下する

ゲームの中で、いろいろなものが落下する場合があります。

そのような時には、  
どのようなプログラムなのか考えてみましょう。

- (1) ネコが上から落ちてきます。
- (2) 雷(かみない)が何度も落ちてきます。

## (1) ネコが落下する

ネコが、上の方から、足をバタバタさせて落ちてくるプログラムです。

### プログラムの説明

- ① いつもの最初の命令です。
- ② X座標が0、Y座標が130ですから、画面の真中より少し上にネコがいます。
- ③ ネコの位置のY座標が“-80”になるまで、④と⑤の命令をくり返す事になります。
- ④ ネコのY座標を“-3”ずつ変えるとは、“3”ずつ下の方向へ降りていきます。
- ⑤ これは、いつもの命令で、ネコが歩いているように見えますね。



## (2) 雷(かみない)が何度も落ちる

### プログラムの説明

- ① いつもの最初の命令です。
- ③ 雷の最初の位置は、X座標が0、Y座標が130です。
- ④ この命令で、雷が表示されます。
- ⑤ この命令で、⑥⑦⑧の命令が40回くり返されます。
- ⑥ もし、雷のY座標が“-240”より小さくなった時(つまり、-240より下に行った場合)、⑦の命令が動作し、でなければ(つまり、-240より上に行った場合)は、⑧の命令が動作します。
- ⑦ 雷が「隠す」とは、雷がぱっと消えます。
- ⑧ 「Y座標を(-10)ずつ変える」とは、この命令を1回実行するごとに、Y座標が“-10”されます。つまり、-10ですから、10だけ下の方に雷が移動します。( -10を40回くり返す前に、雷は-240より小さくなりますから、⑦「隠す」が動作し雷は消えてしまいます。)
- ② 「ずっと」命令で、③から⑧の命令をかこんでいますから、雷が落ちることを何度もくり返します。



(おわり)

# 9 質問とその答え

## 質問とその答え

プログラムの中で、次のようなストーリーが時々起きるものです。

- 1 ある登場人物が質問をする。
- 2 その答えを入力する。
- 3 その入力された答えを、プログラムの中で使用する。

その方法を説明しましょう。

# 動作とそのプログラムの説明

- (1) ①はいつもの開始の命令です。
- (2) ②は、登場人物が話す、**質問の言葉を設定**(せってい)します。
- (3) すると、下のようになり、青矢印の所へ**回答の言葉**(ここでは、ネコの名前: **ミミ**)を入力 します。



- (4) 入力された言葉(ここでは、ネコの名前: **ミミ**)を入力後、**赤矢印をクリック**します。
- (5) クリック後、入力された言葉(ネコの名前: **ミミ**)が黒矢印で示した「**答え**」の中に**登録**(とうろく)されます。
- (6) ③の命令は、ネコが**最後に自分の名前を話す命令**です。「**答え**」にはネコの名前(**ミミ**)が入っていましたね。それと「**だよ!**」とを合わせて話します。



(おわり)

## 10 上下左右キーで動かす

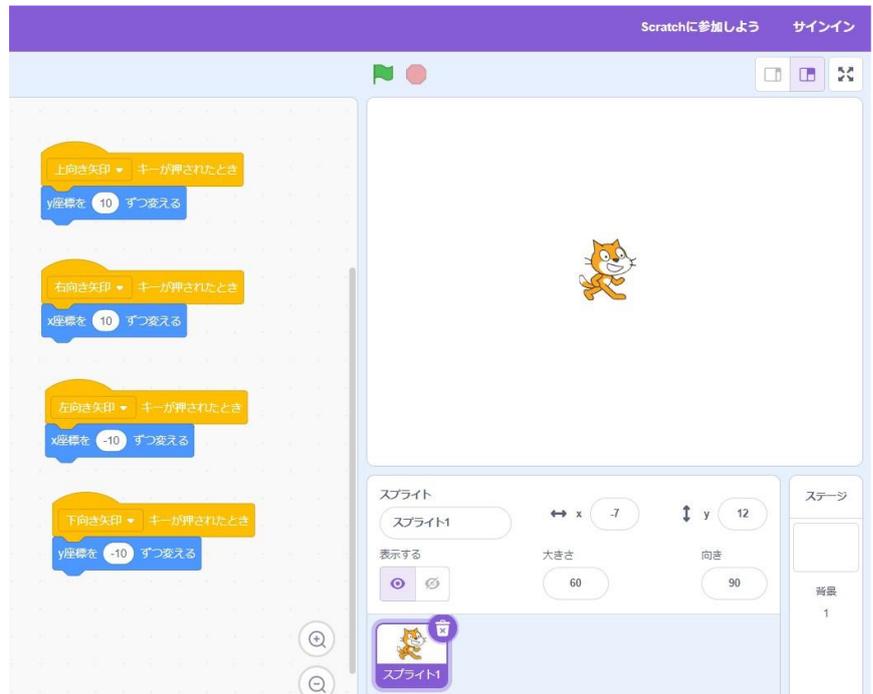
**登場人物(とうじょうじんぶつ)などを、上下左右キーを押すことにより、自由に動かすことができるプログラムを説明しましょう。**

**又、あるキーを押すことにより、いろいろな動作をさせることができる命令について、説明しましょう。**

## 1 上下左右キーで登場人物を動かす

上下左右キーで登場人物を、上下左右に動かすプログラムです。

右のように、  
**上向きキー**  
**右向きキー**  
**左向きキー**  
**下向きキー**  
の4つの命令を  
ならべて書きます。



## 2 各々の命令を、少しくわしく見てみましょう。

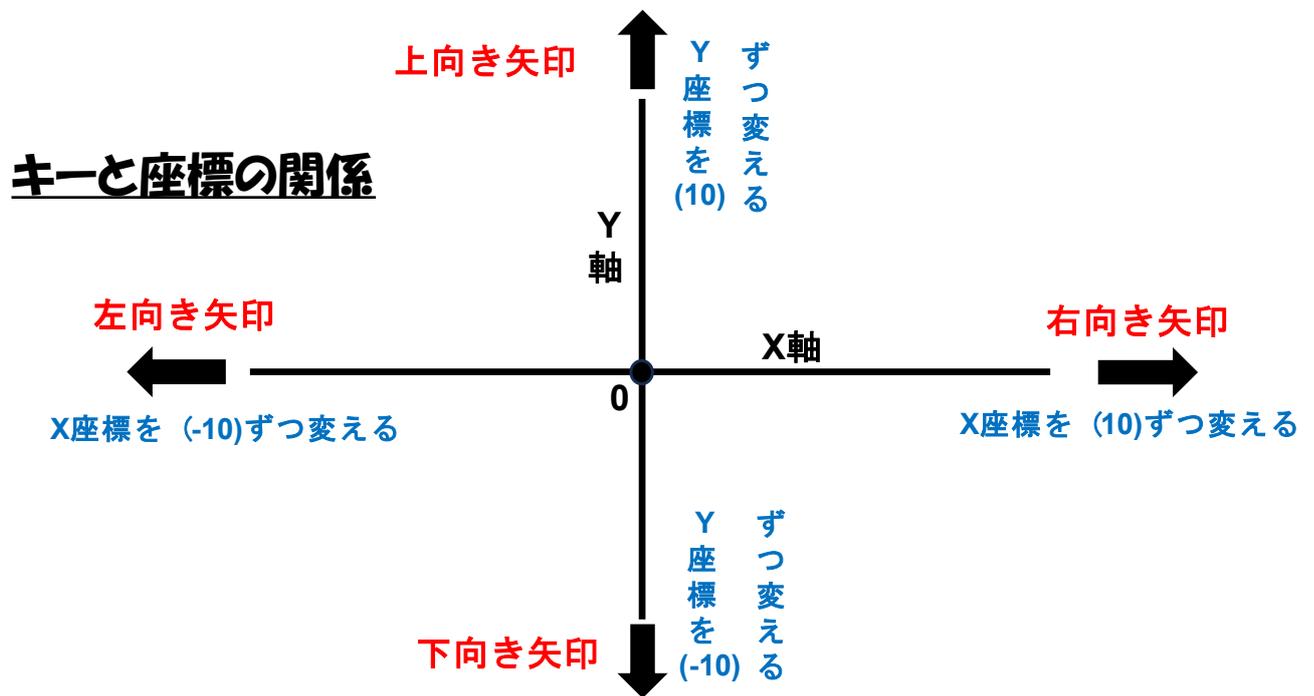


この命令の“スペース”の右にある▼を押すと、キーの一らん表が出てきます。この中から、上向きや右向きなどを選んでください。

- 上向き矢印** ⇒ **Y座標を(10)ずつ変えます。**
- 右向き矢印** ⇒ **X座標を(10)ずつ変えます。**
- 左向き矢印** ⇒ **X座標を(-10)ずつ変えます。**
- 下向き矢印** ⇒ **Y座標を(-10)ずつ変えます。**

キーと座標の関係を次のページに説明しました。





### 3 いろいろなキーを設定(せってい)する。



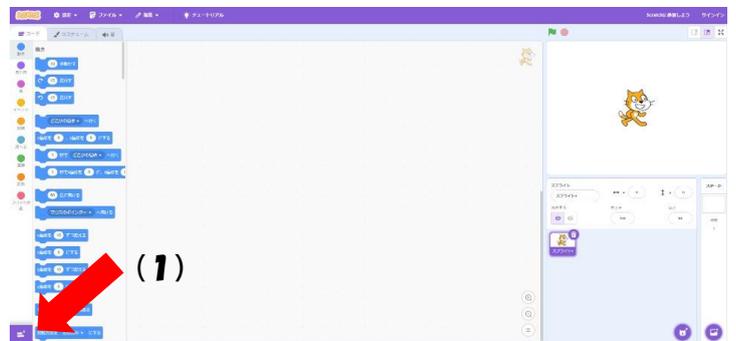
この命令を使うと、上下左右キーだけではなく、いろいろなキーを設定(せってい)して、そのキーが押された時に、いろいろな動作をするプログラムを作ることができます。

(おわり)



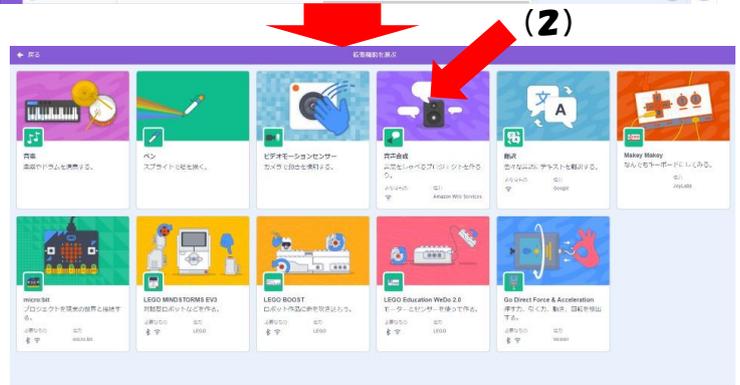
# 11 音声合成を使う

**音声合成**を使うと、プログラム作品  
が**とても分かりやす**くなります。  
簡単ですから使ってみましょう

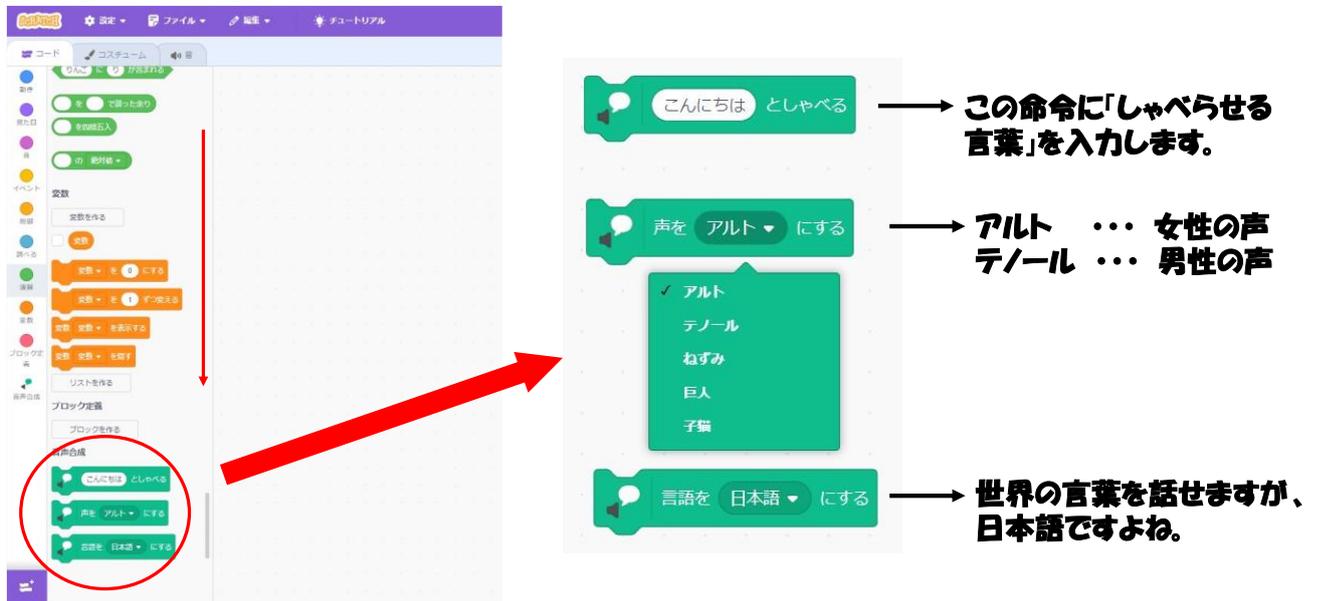


(1) **拡張機能**をクリック

(2) **音声合成**をクリック



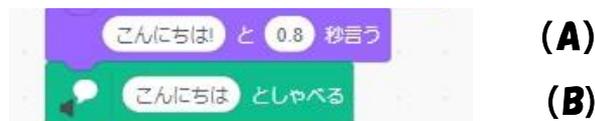
### (3) 音声合成の命令が追加されます。



### (4) 音声合成の命令の使い方

作品の中で「音声合成」を使用すると、とても分かりやすい作品となります。

#### (実際のプログラム例)



このプログラムを動かすと、先ず(A)のお話をする命令が実行され、「こんにちは」と、**0.8秒間風船の中に「こんにちは」と表示**されます。その後、(B)の音声合成の命令が働き、「**こんにちは**」と言います。

(A)(B)と2重に命令をなぜ重ねるかと言いますと、作品の中で登場人物が一人だけだと必要ないのですが、複数の登場人物がいる場合には、誰が話をしているか分からないため、先ず(A)で文字を表示し、その後(B)で言葉をしゃべるのです。

(おわり)

# 12 Scratch作品作成の為に

この作品は、簡単で分かりやすかったですか。

この作品で使ったテクニックを知れば、いろいろな作品を更に作ることが出来そうだと思いますか。

それでは、この作品で使った、主な5つのテクニックについて説明しましょう。

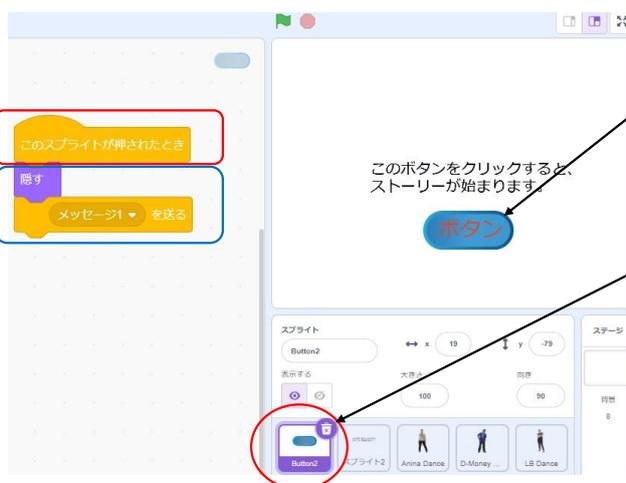
## 1 「ボタンを押したら・・・」

例えば3択(たく)の問題など、ボタンを押したら次の動作に移る場合などが考えられます。

(2) 命令は、「このスプライトが押されたとき」を使います。

(3) この命令の下に、ボタンが押された時に、動作する命令が続きます。

ここでは、「隠す」と「メッセージ1を送る。」の命令が実行されます。



この作品では、このボタンを押す(クリック)すると、作品が始まります。

(1) 登場人物としてボタンのマークを作ります。

## 2 「メッセージを送る」「メッセージを受け取った時」

この作品を進めているのは、「メッセージを送る」「メッセージを受け取った時」の1セットの命令の組み合わせを次々に使い、作品を進めています。この作品では、メッセージ1からメッセージ11まで使ってお話を進めています。

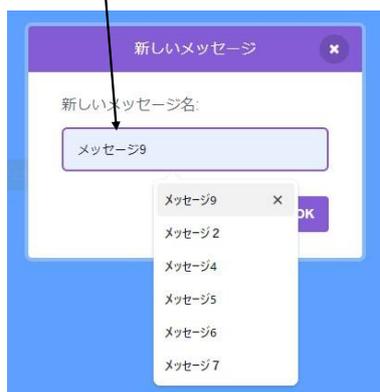


### 「新しいメッセージの作り方」

新しいメッセージをクリックします。



新しいメッセージをクリック名を入力します。



「OK」をクリックします。



### 3 「背景を変える」

作品の進行に従って、背景を変えていく方法を説明しましょう。

命令は、「見た目」にある次の命令のいずれかを使います。



背景の名前は次のページを参照

### 「登録されている背景をみる」

(1) 「背景を選ぶ」をクリック

(2) 「描く」をクリック

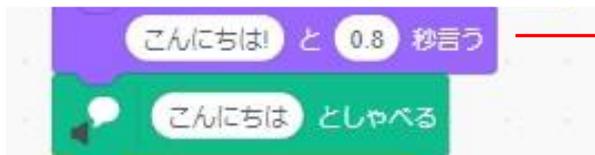
(3) 「背景」をクリック

(4) これが背景の名前です。

## 4 「音声合成」を使用する。

作品の中で「音声合成」を使用すると、とても分かりやすい作品となります。下のプログラムのように「(こんにちは)と(0.8)秒言う」の命令を実行すると、右下の図のように、女の子が「こんにちは」と表示されます。その後に音声合成で「こんにちは」としゃべります。

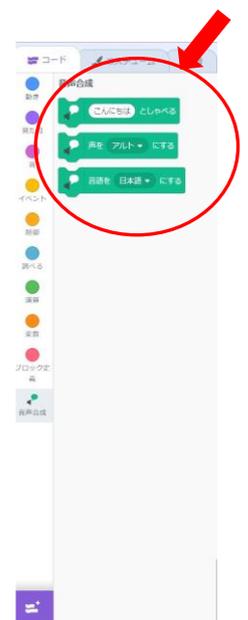
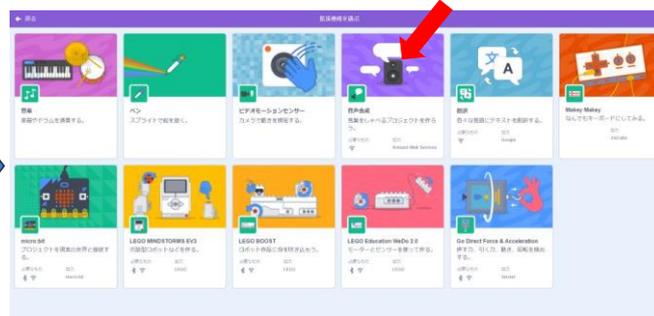
言葉を表示する命令です。  
(0.8秒は「こんにちは」としゃべる大体の時間です。)



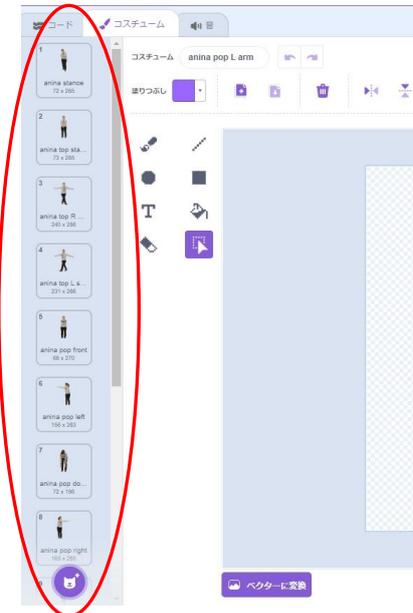
「音声合成」でしゃべる命令です。



## 「音声合成」の設定



## 5 登場人物のコスチュームの指定



登場人物によっては、左のように多くのコスチュームを持っている場合があります。

その際、**コスチュームの指定の方法**について説明しましょう。

### 「コスチューム」の設定



**この命令を使用！**

