

上 級

5 基本的動き方を学ぶ。

上級の初めに、座標の考え方と登場人物の基本的な動かし方を学びましょう。

(5-1) 座標について学ぶ。

(1)Scratch でプログラミング作品を作る際に、どうしても学んでおかなければならない、「座標」について学びましょう。

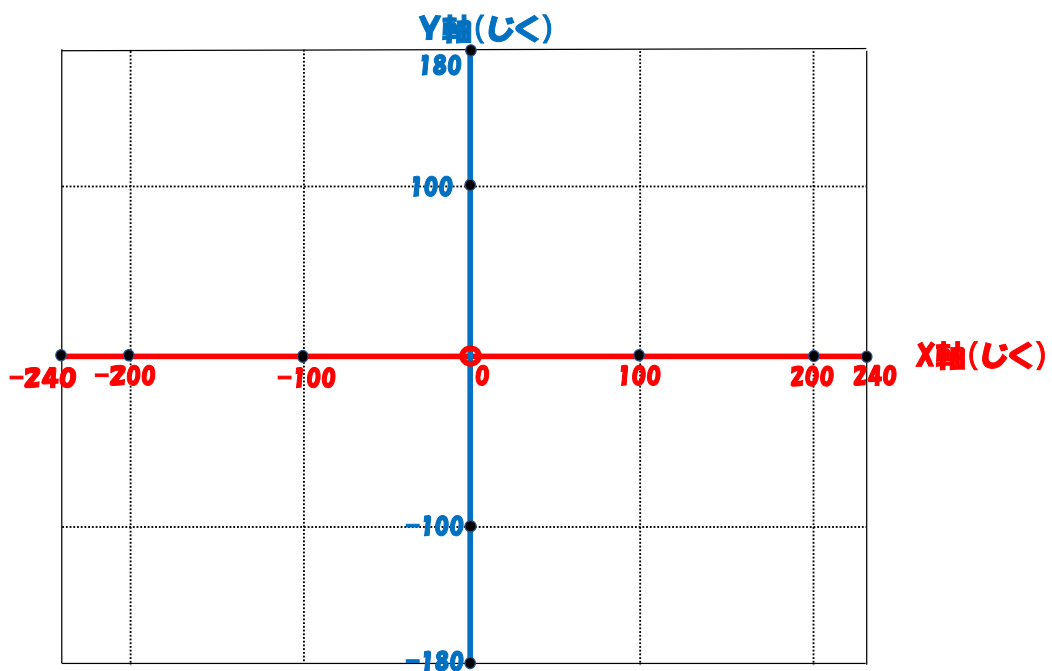
そこで、先ず X 軸や Y 軸など座標の概念について説明します。

Scratch では、(図5-1)のように、

X軸 : -240~240

Y軸 : -180~180

の値を取ることが出来ます。

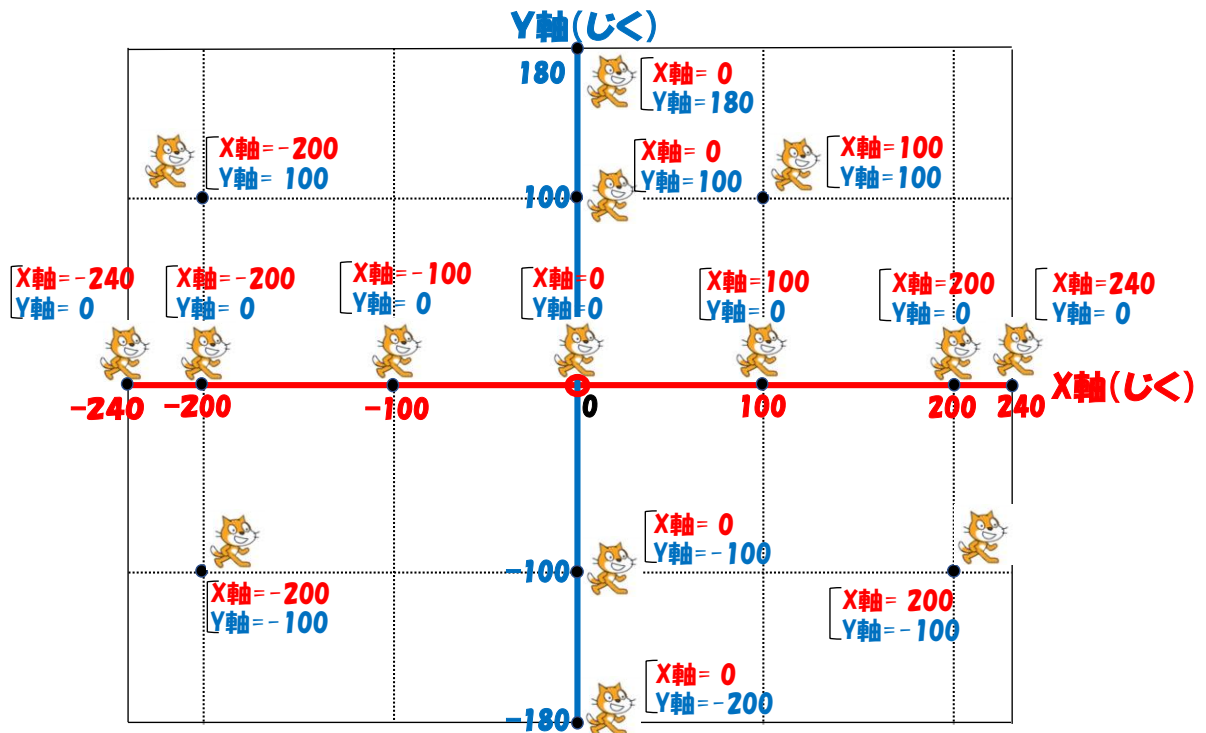


(図5-1)

(2)次に(図5-2)のように、いろいろな位置にいるネコの座標を、児童に答えてもらいます。

コメント

4年生でも、座標について理解し、次々にネコの座標を答える事が出来るものです。この座標のP.P.は、補助教材の中にあります。



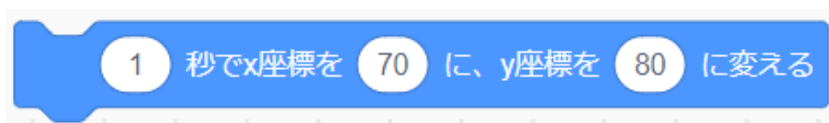
(図5-2)

(3)この座標の考えを使って、作品を作るために使う命令は次の2つです。

登場人物の位置を、 $X=50$ 、 $Y=60$ にします。



登場人物の位置を、1秒後に $X=70$ 、 $Y=80$ にします。



(5-2) 登場人物を上下左右に動かす。

登場人物を上下左右に動かす場合には、一般的には上向き矢印・下向き矢印・左向き矢印・右向き矢印で、上下左右に動くように設定します。

(1) 上下左右ボタンの設定

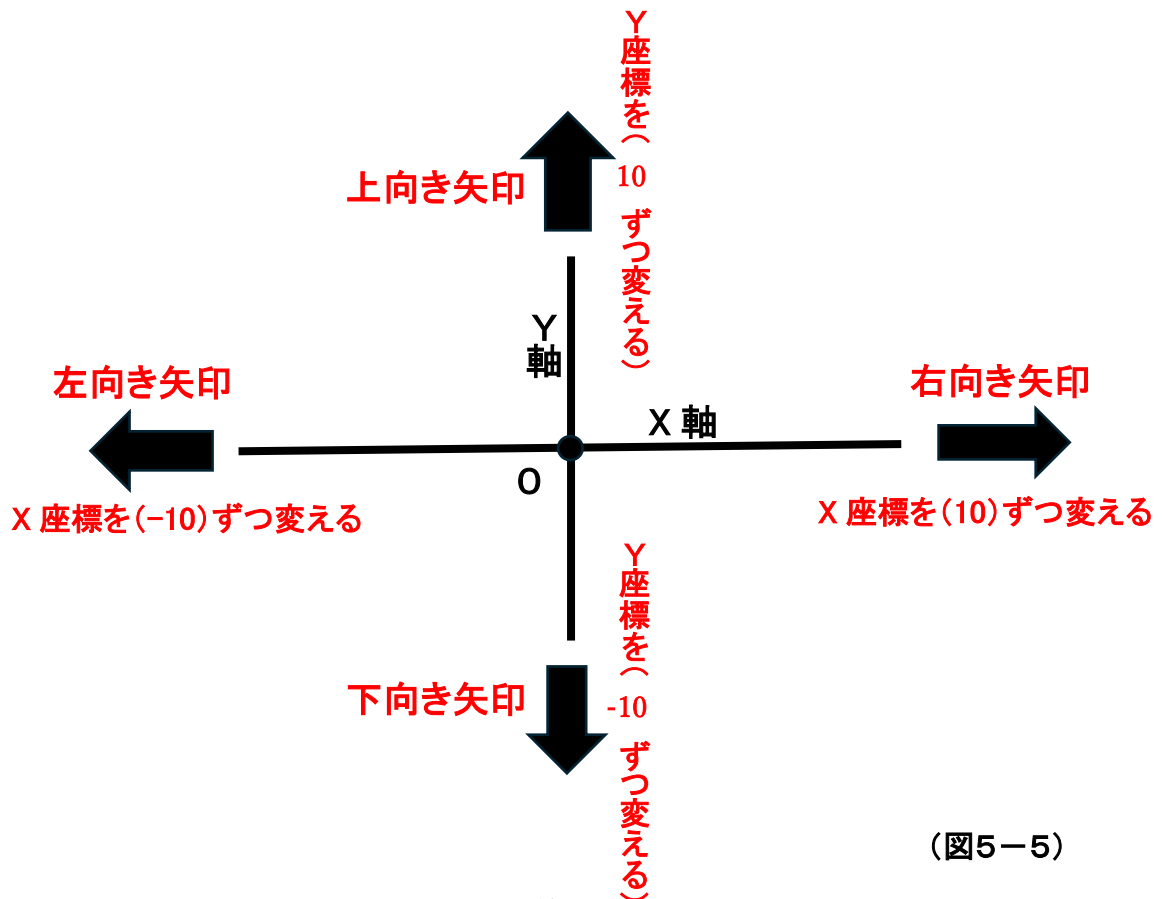


(図5-3)



(図5-4)

(2) 上下左右に動かす基本的考え方



(図5-5)

(3) 登場人物を上下左右に動かすプログラム



(図5-6)

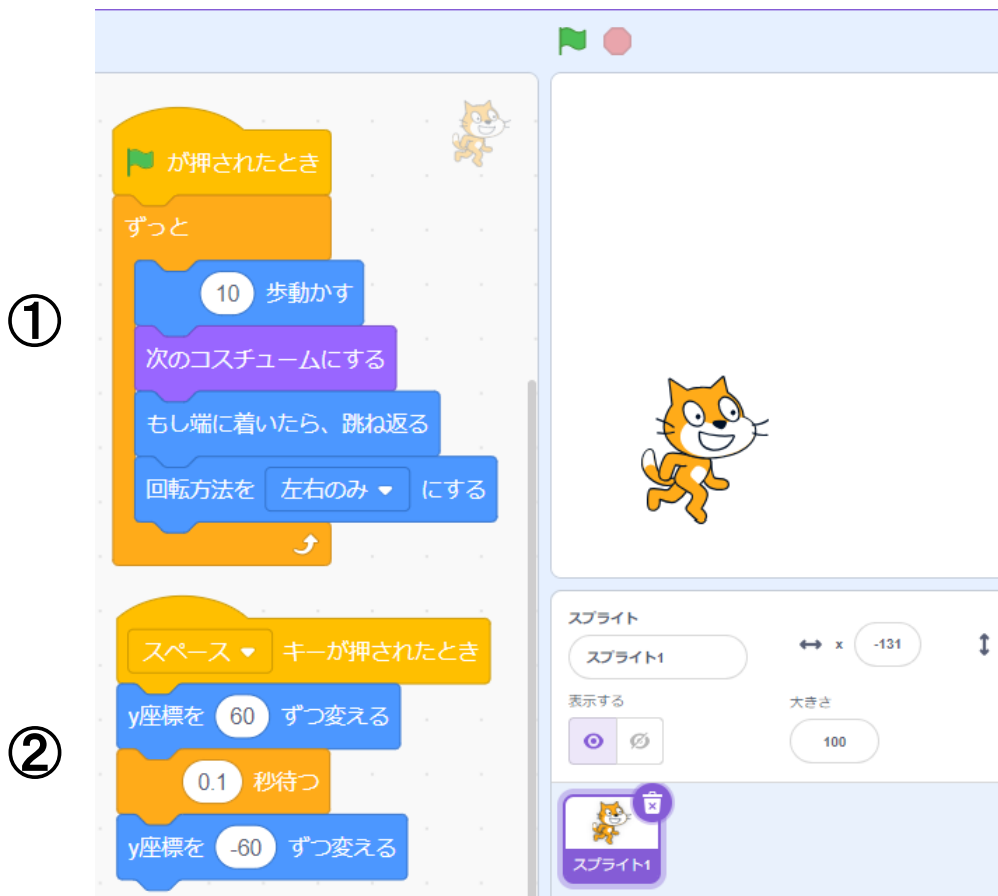
コメント

座標や上下左右に動かすは、基本的操作です。
確実に覚えましょう。

(5-3) ネコのジャンプ。

ゲームなどでは、主人公を「ジャンプ」動作させることがあります。
2種類の「ジャンプ」を説明しましょう。

(1) スペースボタンを押し、ネコをジャンプさせよう。



(図5-7)

- 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、もう何度も勉強してきました。
- ② このジャンプするプログラムの3つの命令と、ネコの動きの関係は次の通り。



(図5-8)

(参考) ネコに緑の階段を上り下りさせよう。

スペースボタンを押すと、ネコがジャンプし緑の階段に上り、又その階段を下りるプログラムです。

コメント

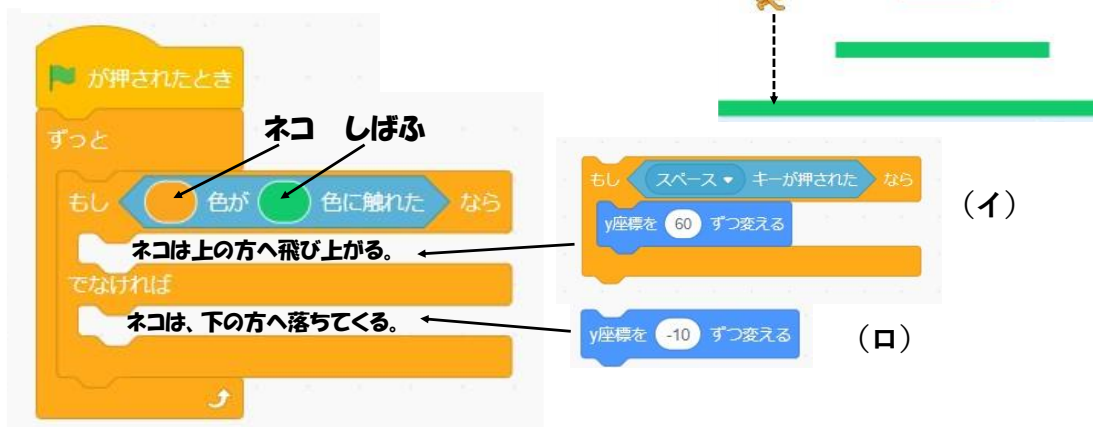
少し理解が難しいかもしれませんが、時間が余ったら学びましょう。



① ネコが左右に歩き続けるプログラム。

(図5-9)

② スペースボタンを押すと、ネコが飛び上がるプログラム。



(図5-10)

(5-4) 今日習った、座標やネコを上下左右に動かす、又はネコをジャンプさせるプログラムを使って自分のプログラムを作ってみよう。

(座標)

x座標を 50 、y座標を 60 にする

1 秒でx座標を 70 に、y座標を 80 に変える

(上下左右)

上向き矢印 ▼ キーが押されたとき

y座標を 10 ずつ変える

下向き矢印 ▼ キーが押されたとき

y座標を -10 ずつ変える

左向き矢印 ▼ キーが押されたとき

x座標を -10 ずつ変える

右向き矢印 ▼ キーが押されたとき

x座標を 10 ずつ変える

(ネコのジャンプ)

スペース ▼ キーが押されたとき

y座標を 60 ずつ変える

0.1 秒待つ

y座標を -60 ずつ変える

が押されたとき

ずっと

もし 色が 緑 に触れた なら

もし スペース ▼ キーが押された なら

y座標を 60 ずつ変える

でなければ

y座標を -10 ずつ変える

(図5-11)

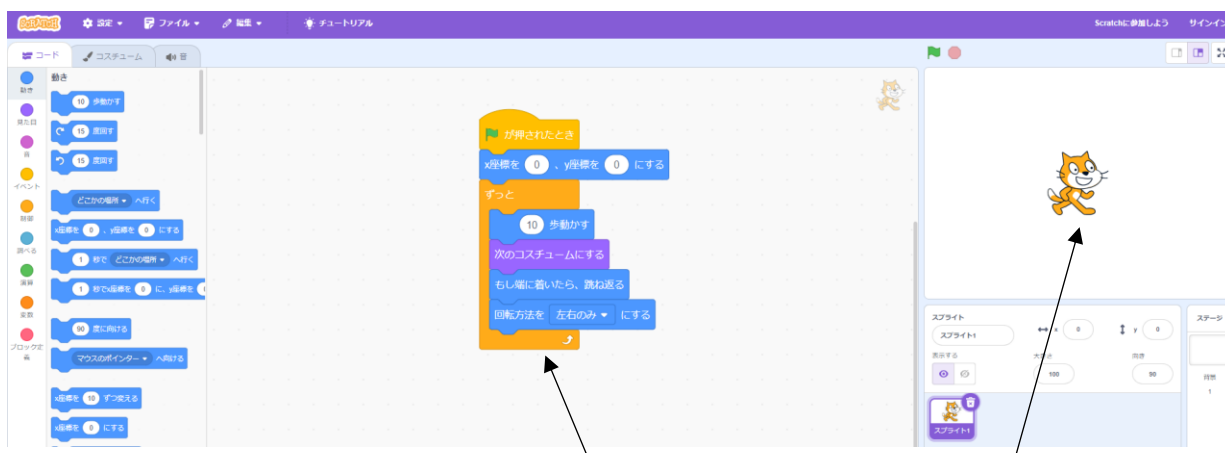
2時間目

6 プログラムを作る為の考え方を学ぶ。

これから児童は、自分で創造的プログラム作品を作らねばなりません。そのプログラムを作る為に、どのように考えながら作るのか説明しましょう。

(6-1) 先ず初めに。

先ず初めに、このネコが動くプログラムを作る場合、どのように考えますか。



(図6-1)

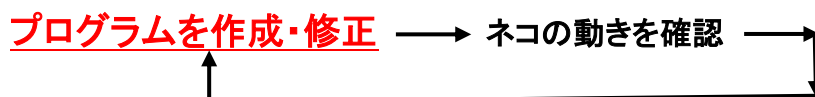
プログラム

ネコの動き

プログラムを作る時に、プログラムをあまり見ずにネコの動きを見ながら、一つずつ命令を加えてプログラムを作っていないか。

この方法では、遅かれ早かれプログラミングに行き詰まってしまう。

重要なことは、プログラムを常に考えながら作り、大体プログラムが出来たらプログラムを動かします。そのネコの動きを確かめて、もし問題があればプログラムを修正します。それを繰り返しプログラムを作っていくことが重要なのです。つまり、常にプログラムを考えながらプログラムを作る事が重要なのです。

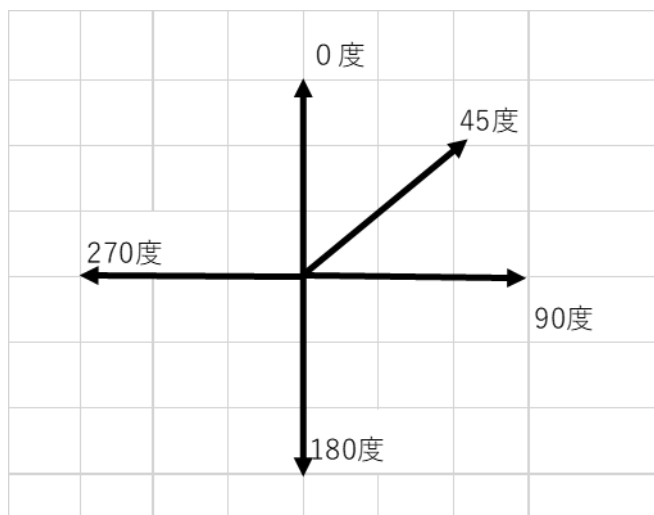


(6-2) その前に、Scratch で重要な決まり事を1つ説明しましょう。

Scratch では下図のように、上向きのことを角度で「0度」と言います。

そうしますと、真右は「90度」と表示しますし、真下は「180度」、真左は「270度」又は「-90度」という事になります。

この角度による方向の表示は、Scratch では、常にこのようになります。



コメント
Scratch の基本的取り
決めです。

(図6-2)

(6-3) 今回は「ネコが円を描いて動く」というプログラムを使って、プログラムはどのように動いているのか理解しましょう。

(図6-3)は、ネコが円を描くプログラムです。プログラムは、コンピューターが最初の命令から次の命令へ順番に実行し、ネコを動かしていきます。このプログラムでネコがどのように動くかを理解する為に、命令と命令の間に([2秒待つ])の命令を4個挿入します。これで、プログラムの働きは変えずに、次の命令を実行する前に2秒ずつ休止を作る事になります。これをジックリと見ると、各々の命令の働きが分かります。それでは、([2秒待つ])の命令を挿入したプログラムも並べて(図6-4)に示しました。

この([2秒待つ])の命令を挿入する方法は、プログラムをデバックする場合にも使用することが出来ます。

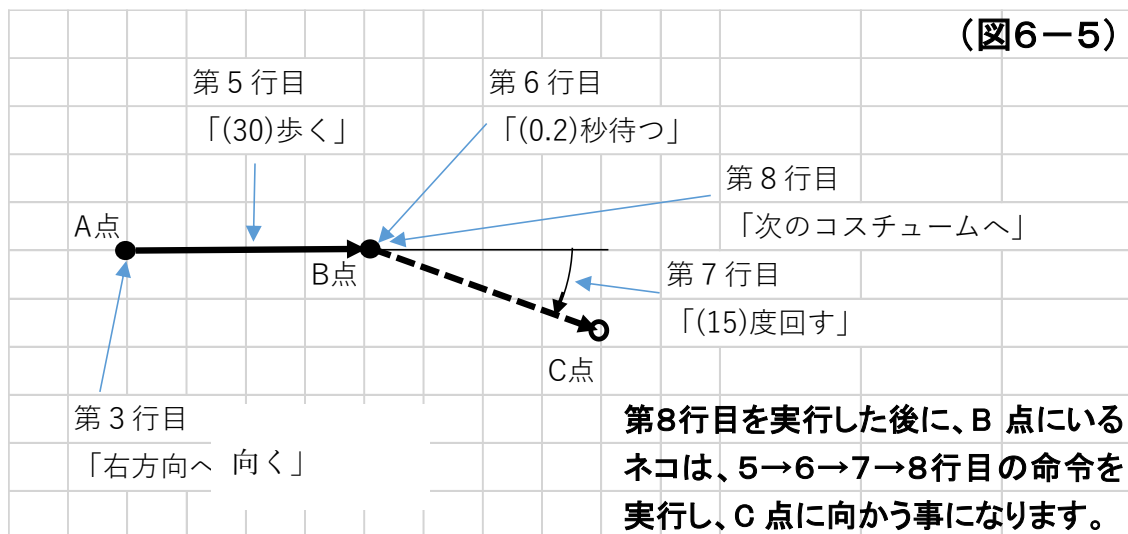


(図6-3)



(図6-4)

(図6-4)のプログラムの動作をじっくりと見た結果、このプログラムとネコの動きとの関係は、次のようになっている事が確認できるはずです。



又、これにより、ネコが円を描くプログラムは、実際は円ではなく多角形を描いていたことが分かります。

さて、今回のネコが円を描くプログラムで、各々の命令がどのような働きをして、円を描いたか理解出来たのではと思います。

コンピューターは、このようにプログラムを実に忠実に最初から一歩一歩実行して行くだけなのです。プログラム作者の意図と違った動きをしても、それはほとんどの場合、プログラムのミスか十分に考えられていなかったからなのです。

実は児童がプログラム作品を作るという事は、プログラムとともに登場人物の動きも考えてプログラムを作る必要があります。何となくプログラムをコピーして、うまく動かない時などバグを修正する必要がありますが、このような**一つ一つの動作まで踏み込んで考えないと、バグの修正が出来ない**ことがあります。

コメント

プログラムとは、このように一つ一つ分解すると、どんなプログラムでも単純な命令の動きの積み重ねから出来ています。バグを探したり、修正する時には参考にしましょう。

(6-4) 「ネコの宙返り」のプログラムを使い、コンピューターの基本的動きを知る。

このプログラムは、ネコの回転する命令[90度回す]とその命令を繰り返す命令[20回繰り返す]だけの単純なプログラムです。このプログラムを参考にし、Scratch の回転する命令の基本的動きを確認しましょう。

[0.2秒待つ]命令は、ネコの回転を遅くする為に付け加えられています。

(1) 90度を20回繰り返す



(図6-6)

プログラムは、90度右回転する命令を20回繰り返しています。それでは、このプログラムを動かすと、ネコは何回回転するでしょうか。

(答え)

90度を20回ですから、ネコは5回回転します。実際に先生がプログラムを動かして、皆で確認しましょう。

(2) 90度を8回繰り返す



(図6-7)

プログラムは、90度右回転する命令を8回繰り返しています。それでは、このプログラムを動かすと、ネコは何回回転するでしょうか。

(答え)

90度を8回ですから、ネコは2回回転します。実際に先生がプログラムを動かして、皆で確認しましょう。

(3) 18度を20回繰り返す



(図6-8)

プログラムは、18度右回転する命令を20回繰り返しています。それでは、このプログラムを動かすと、ネコは何回回転するでしょうか。

(答え)

$18\text{度} \times 20\text{回} = 360\text{度}$ ですからネコは1回だけ回転します。実際に先生がプログラムを動かして、皆で確認しましょう。

コメント

この問題は事前に児童に考えさせ、その後プログラムを動かして、結果を考察させる事が出来ます。

(4) 360度を20回繰り返す



(図6-9)

プログラムは、360度右回転する命令を20回繰り返しています。それでは、このプログラムを動かすと、ネコは何回回転するでしょうか。

(答え)

何度やっても、ネコは微動もしません。しかし、実際は、ネコは20回回っているのです。[360 度回す]という命令を実行すると、コンピューターは、目にも止まらない速さで1回回転しますが、高速過ぎて回っているようには見えないのです。

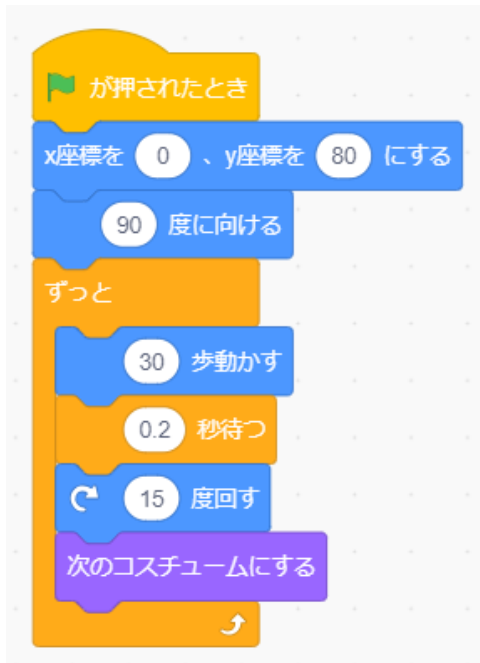
(5) ネコを早く回転させたり、遅く回転させるにはどうすればよいでしょうか。

(答え)

ネコを早く回転するには、[(0.2)秒待つ]命令の(0.2)秒を小さくすればよいですし、逆に遅く回転させるには、(0.2)秒を大きくすればよいのです。

(6-5) 今回習った、ネコを回転させるプログラムを使って、自分のプログラムを作ってみよう。

(ネコが円を描いて動く)



(図6-10)

(ネコの宙返り)



(図6-11)

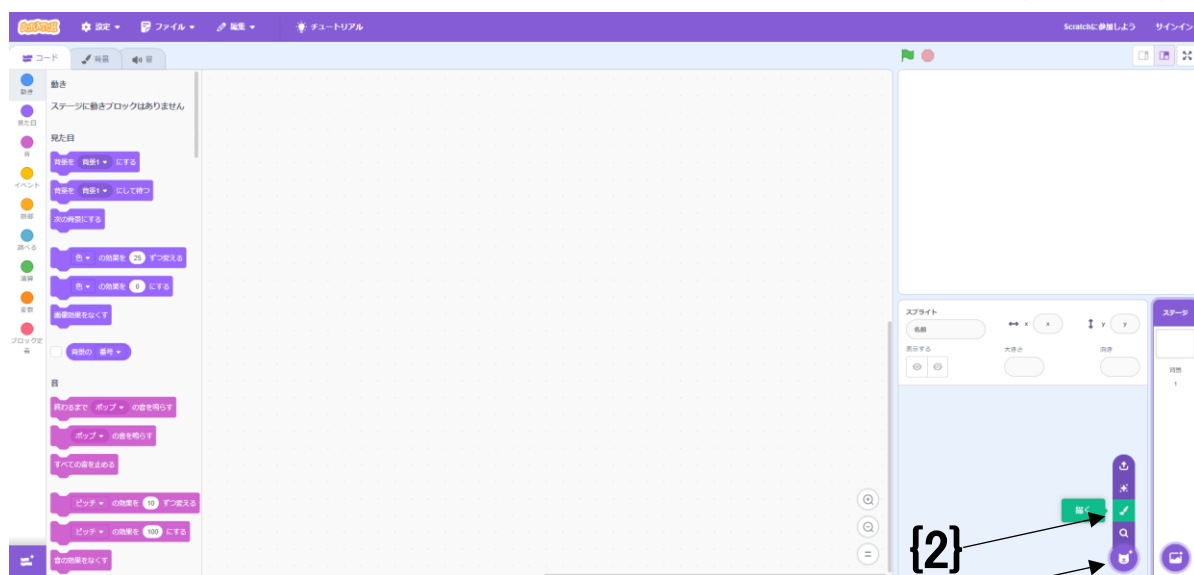
3時間目

7 ボールを壁に当てると、ボールが消える。

ボールを壁に当てるとボールが消えてしまいます。これは、「もし～なら…」という条件文を使うテクニックです。少し複雑なプログラムを作る場合には、必ずと言っていい程必要な、プログラミングの基本テクニックです。是非習得しましょう。

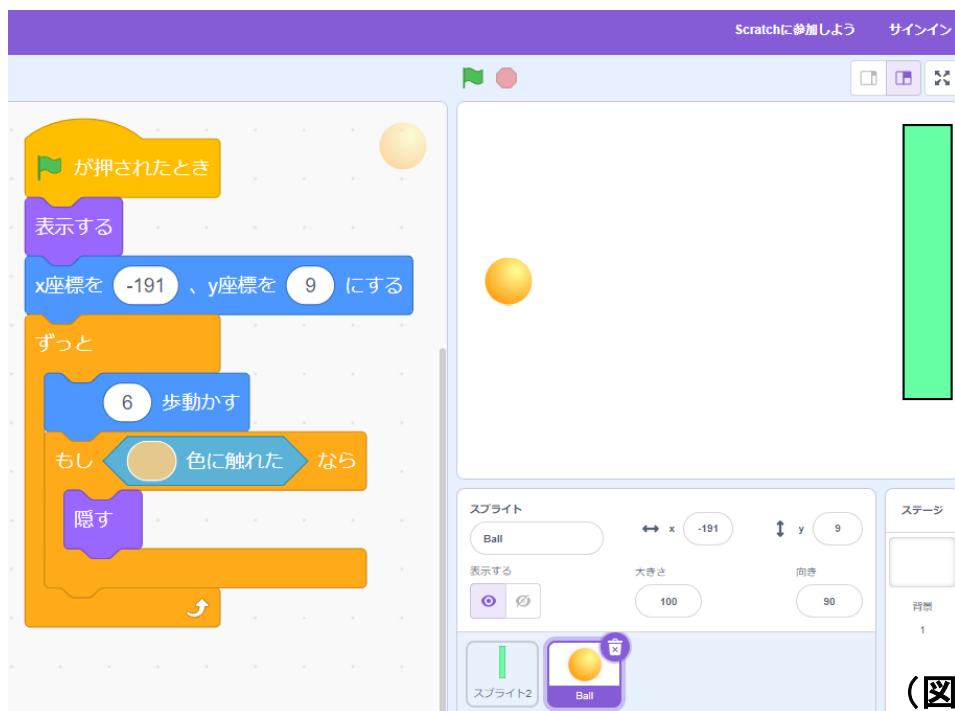
(7-1) 先ず初めに、壁を作りましょう。

(図7-1)

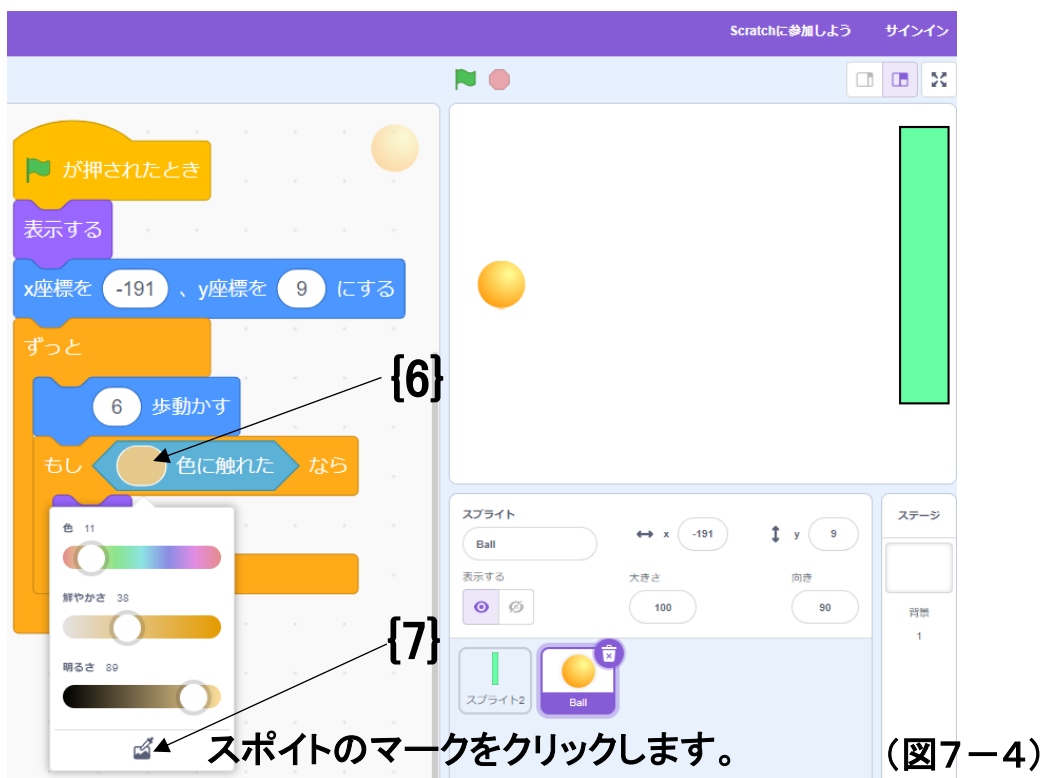


(図7-2)

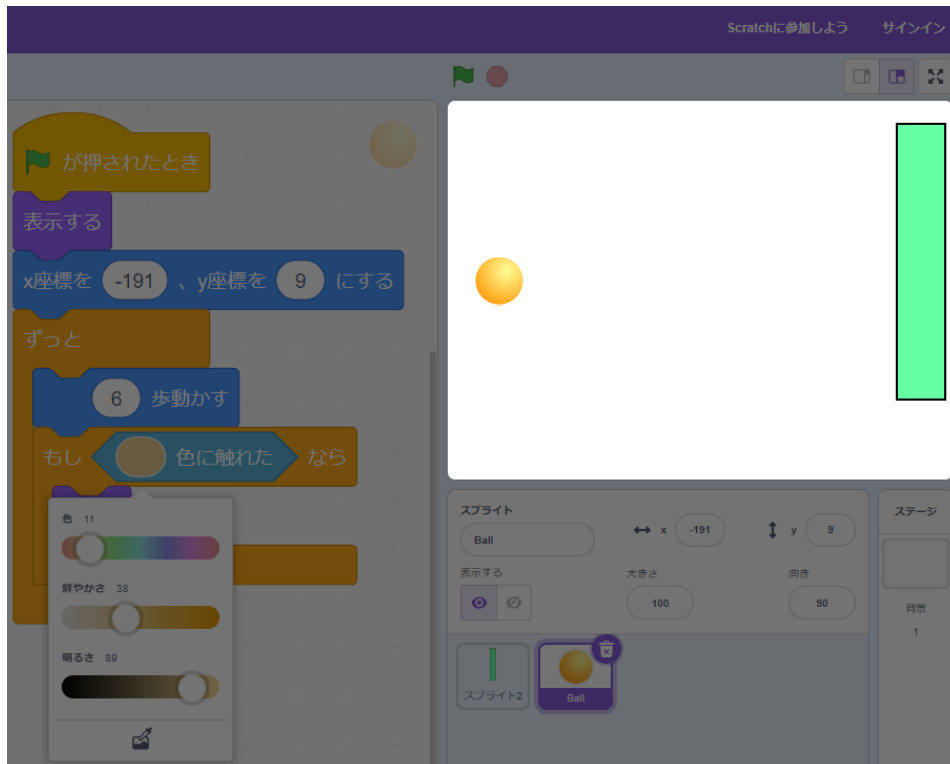
(7-2) 登場人物にボールを追加し、ボールにプログラムを書きましょう。
登場人物としてボールを登録し、下記のプログラムを書きましょう。



緑の壁に当たったらボールが消える為には、壁の緑色の指定が必要です。
その方法は次の通りです。

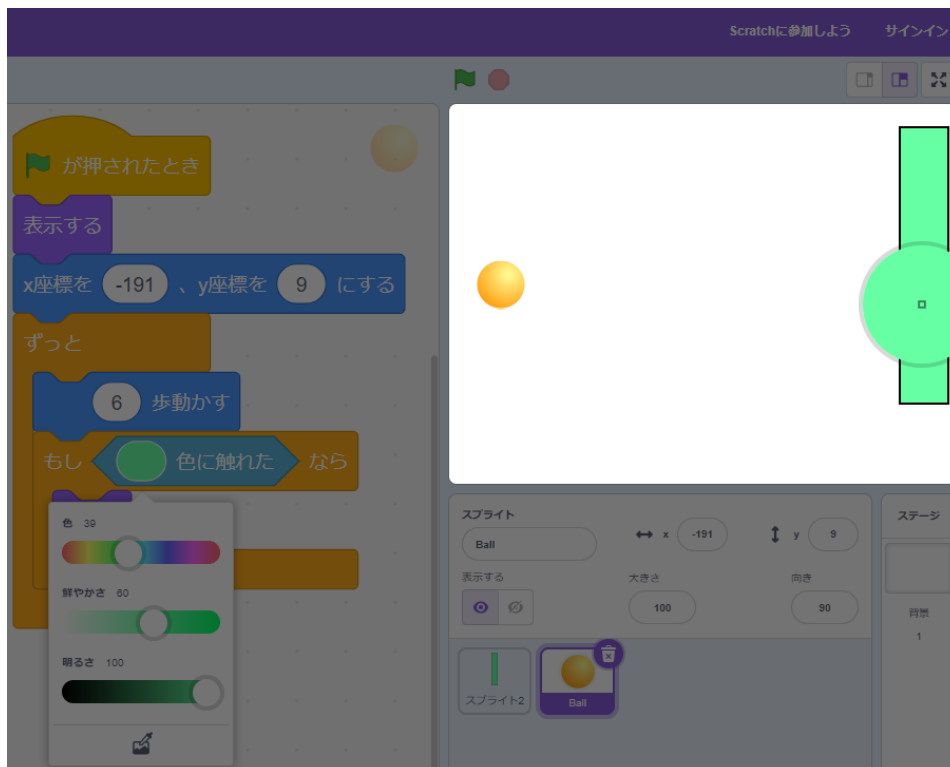


スポイトのマークをクリックすると下記のように画面が変わります。



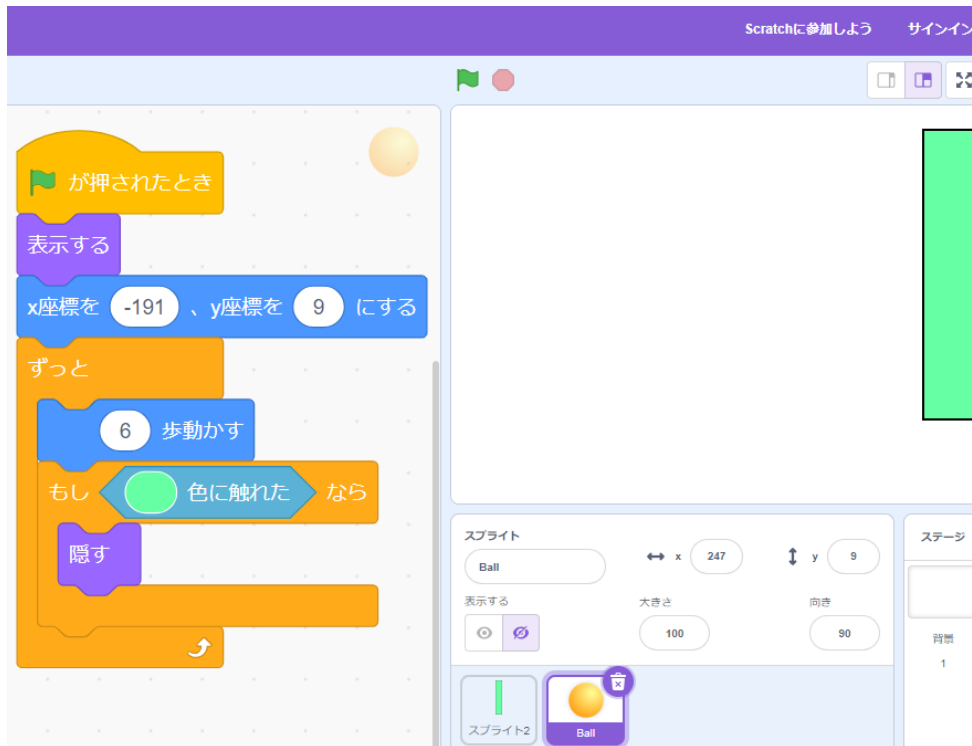
(図7-5)

ポインターを色合わせしたい緑の壁に合わせ、左クリックします。



(図7-6)

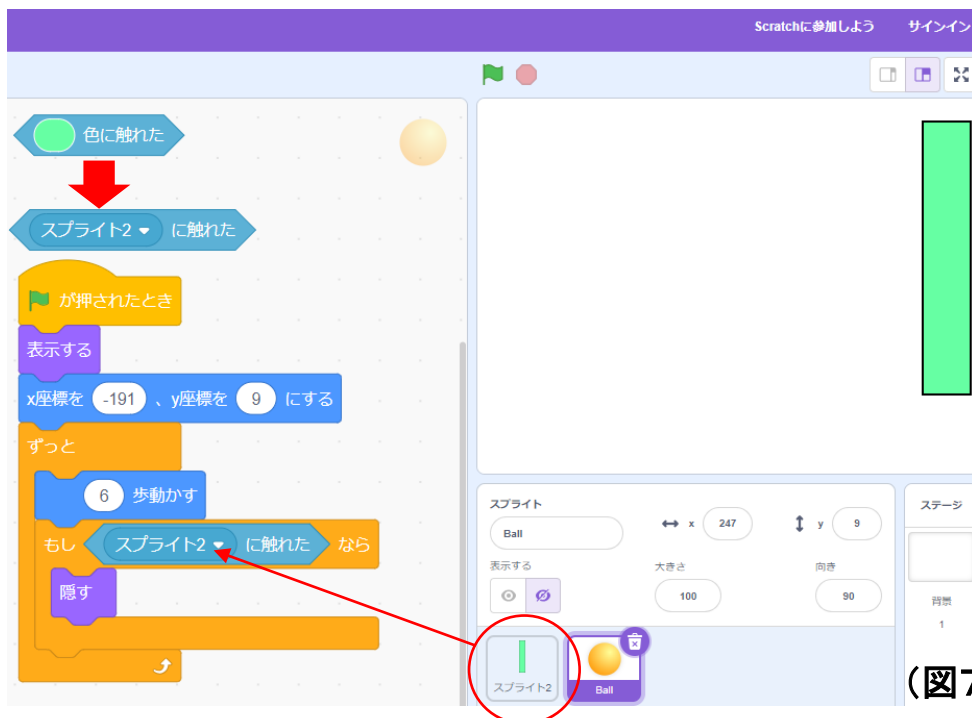
これでプログラムは完成です。緑の旗を何度もクリックし、動作を確認しましょう。



しかしこのプログラムには、幾つかバージョンがあります。 (図7-7)

(バージョン1)

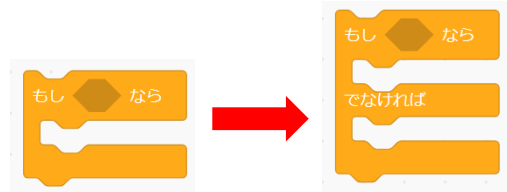
“()色に触れた”命令から“(登場人物)に触れた”命令への変更。



(図7-8)

(バージョン2)

条件文の変更。どちらの命令を使っても同じ動きをします。



(図7-9)

(7-3) 今回習った、「ボールを壁に当てると、ボールが消える。」プログラムを使って自分のプログラムを作ってみましょう。

(ボールを壁に当てると、ボールが消える) <色合わせ方式>



(図7-10)

(ボールを壁に当てると、ボールが消える) <登場人物指定方式>



(図7-10)

(ボールを壁に当てると、ボールが消える) <別の条件文使用>



(図7-11)

8 お話をする。

二人の人がお話をするプログラムです。このプログラムは、お話をするだけでなく、次々に物語等を進めて行くために、是非必要なプログラミングの基本テクニックです。是非習得しましょう。

(8-1) 女の子と男の子に、次のようなプログラムを書きましょう。

女の子と男の子の間には、次のように会話が進みます。

(女の子)“おはよう” → (男の子)“おはよう、ひさしぶりだね” →
(女の子)“何してるの” → (男の子)“なにをして遊ぼうか、考えているんだよ”
→(女の子)“それじゃね、さようなら”

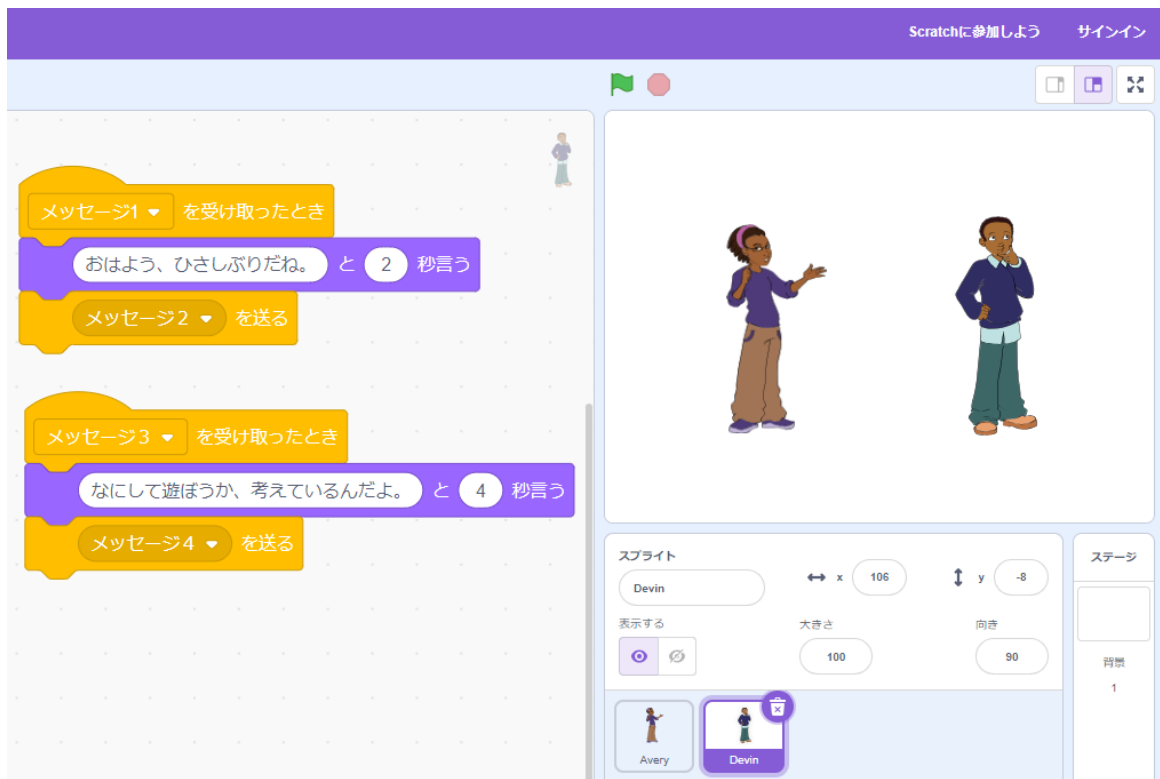
この会話を実現させる為のプログラムは、次の通りです。

(女の子)



(図 8-1)

(男の子)



(図 8-2)

(8-2) このプログラムの仕組みの説明。

この二人の会話を進めているのは、「メッセージを送る」と「メッセージを受け取ったとき」の1セットの命令です。この例では、4セットの命令を次々に使い会話を進めています。

その1セットの命令は下記の通りです。

(A)



(B)



(図 8-3)

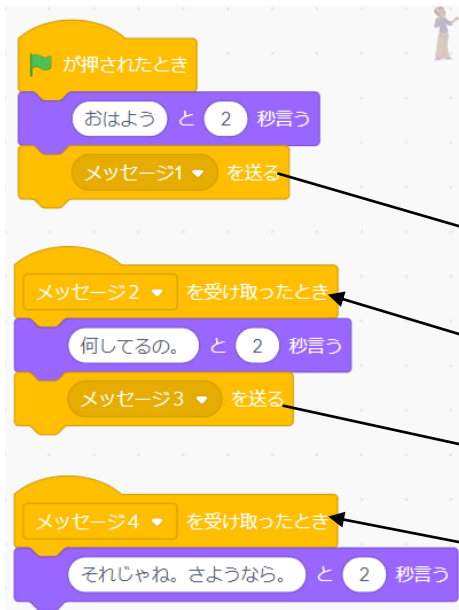
(A)の命令は、「メッセージ1」という**名前**の**信号**を出す命令です。

(B)の命令は、「メッセージ1」という**命令**を**受け取る**命令です。

* 命令の名前は、どのような名前でも使用できます。

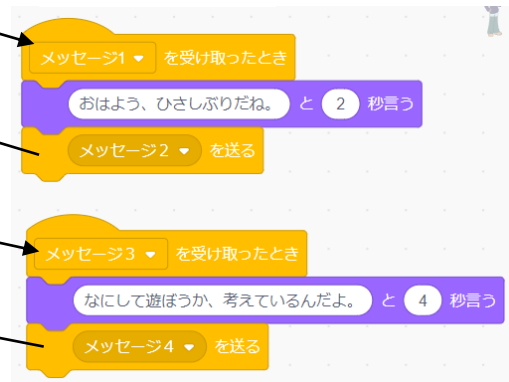
それでは、女の子と男の子のプログラムの命令の流れを見てみましょう。

(女の子)



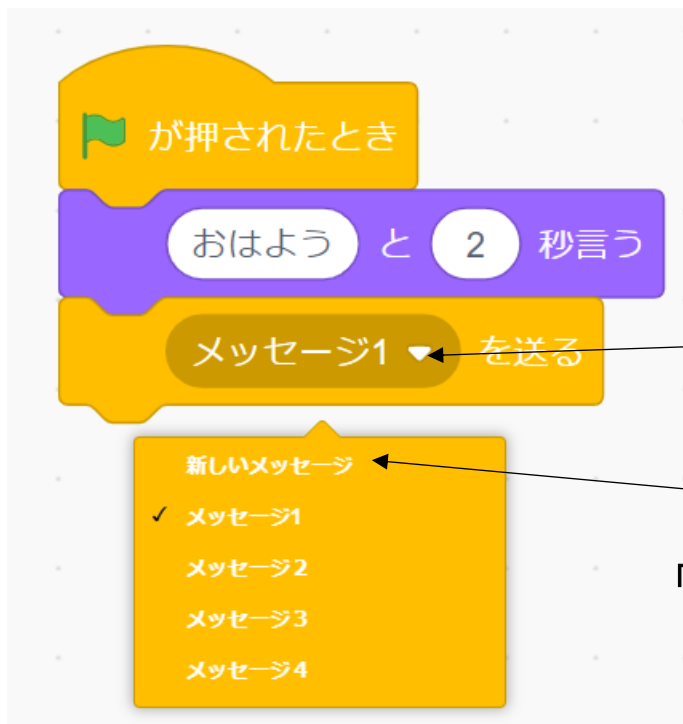
(図 8-4)

(男の子)



(図 8-5)

(8-3) 「メッセージ1」等の信号名を増やす方法。



{1}
「▼」をクリックします。

{2}
「新しいメッセージ」をクリックします。

(図 8-6)



{3}

新しい信号名を入力します。
信号名はどのような名前でも
かまいません。

(図 8-7)

{4}

「OK」をクリックします。

(8-4) お話ではない使い方 (ネコが木から落ち、驚いて鳥が飛び出すお話)

次に物語を進める為に、このテクニックを使用した例を説明しましょう。

このお話は、「ネコが木を登っているうちに落ちてしまいました。その木の下には小鳥と鶏が遊んでいますが、ネコが落ちてきて驚いて飛び出していきます。」

このお話の中で、ネコは木から落ちてきて地面にぶつかった時に、「メッセージ1」の信号を出します。すると小鳥と鶏は「メッセージ1の信号を受け取った」後に、飛び出します。

それでは、先ず「ネコのプログラム」「鳥のプログラム」「鶏のプログラム」の全体を示します。

コメント

このプログラムは、補助教材の中にあります。
使用しましょう。

(ネコのプログラム)

(図 8-8)

Scratch code editor showing a program for a cat. The code is organized into two main sections triggered by a green flag click.

Section 1 (Left):

- 当緑旗がクリックされたとき
- 大きさを 70 %にする
- x座標を 143 、y座標を 137 にする
- 120 度に向ける
- 10 回繰り返す
 - 0.3 秒待つ
 - 10 歩動かす
 - 次のコスチュームにする
- 29 回繰り返す
 - 0.2 秒待つ
 - y座標を -10 ずつ変える
 - 90 度回す
- メッセージ1 を送る

Section 2 (Right):

- 当緑旗がクリックされたとき
- 2 回繰り返す
 - 終わるまで Drum Roll の音を鳴らす
- 12 回繰り返す
 - 終わるまで Slide Whistle の音を鳴らす
- 7 回繰り返す
 - 終わるまで Big Boing の音を鳴らす
- 7 回繰り返す
 - 終わるまで Chirp の音を鳴らす

The stage shows a cat on a tree branch, a chick, and a hen. The sprite panel shows the cat as the selected sprite, with Chick and Hen also visible. The stage background is a landscape with a table and chairs.

(小鳥のプログラム)

(図 8-9)

Scratch code editor showing a program for a chick. The code is organized into two main sections.

Section 1 (Left):

- メッセージ1 を受け取ったとき
- 13 回繰り返す
 - コスチュームを chick-b にする
 - 10 歩動かす
 - 0.1 秒待つ
 - コスチュームを chick-c にする
 - 10 歩動かす
 - 0.2 秒待つ

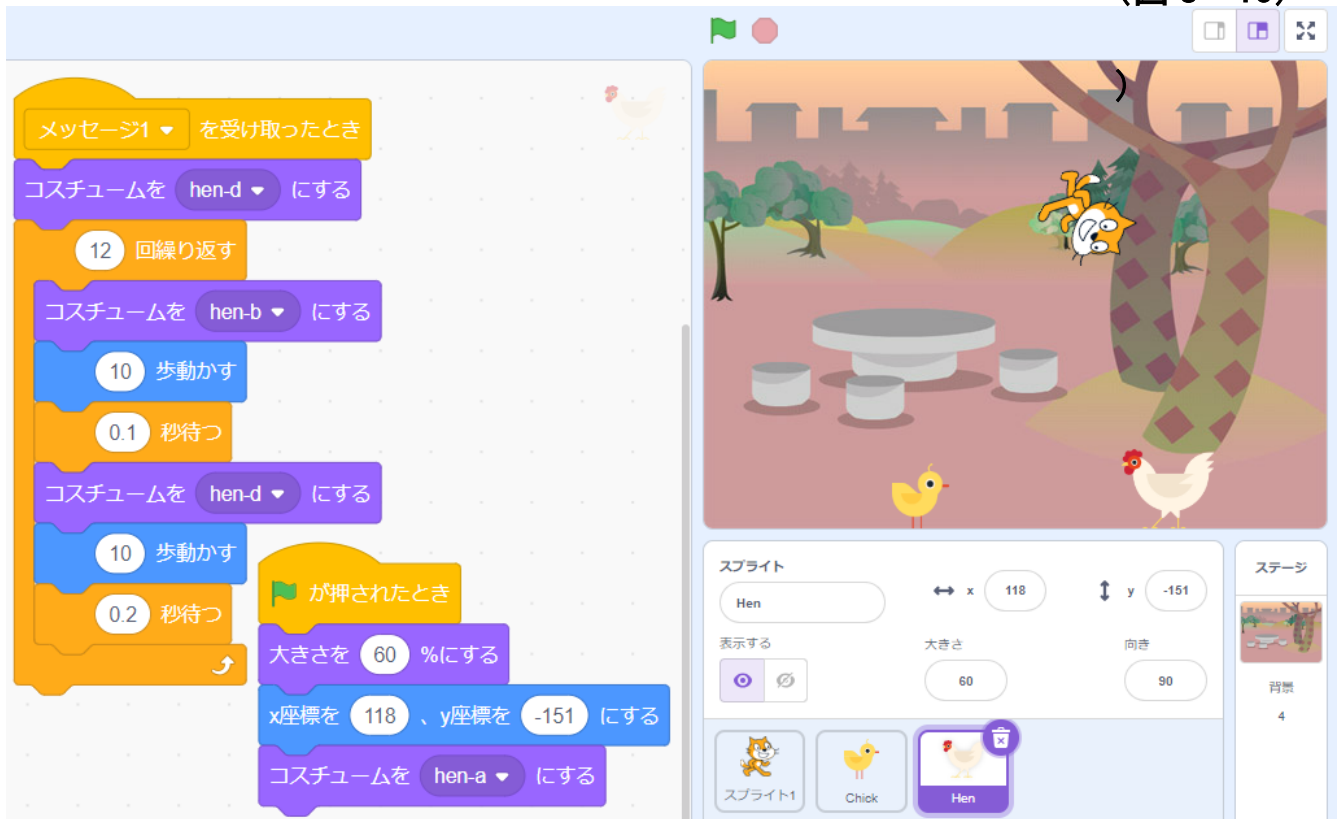
Section 2 (Right):

- 当緑旗がクリックされたとき
- 大きさを 70 %にする
- x座標を -73 、y座標を -156 にする
- コスチュームを chick-a にする

The stage shows the cat on the tree branch, the chick, and the hen. The sprite panel shows the chick as the selected sprite, with the cat and hen also visible. The stage background is the same landscape as in Figure 8-8.

(鶏のプログラム)

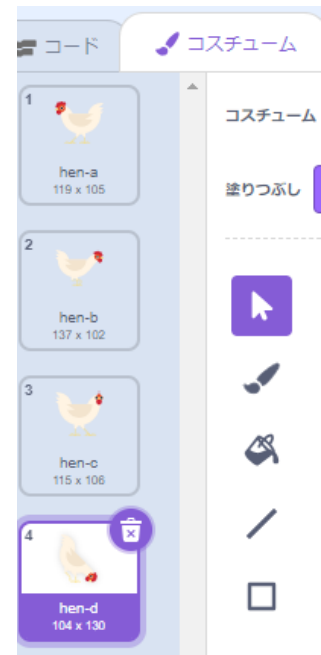
(図 8-10)



(ネコのコスチューム)

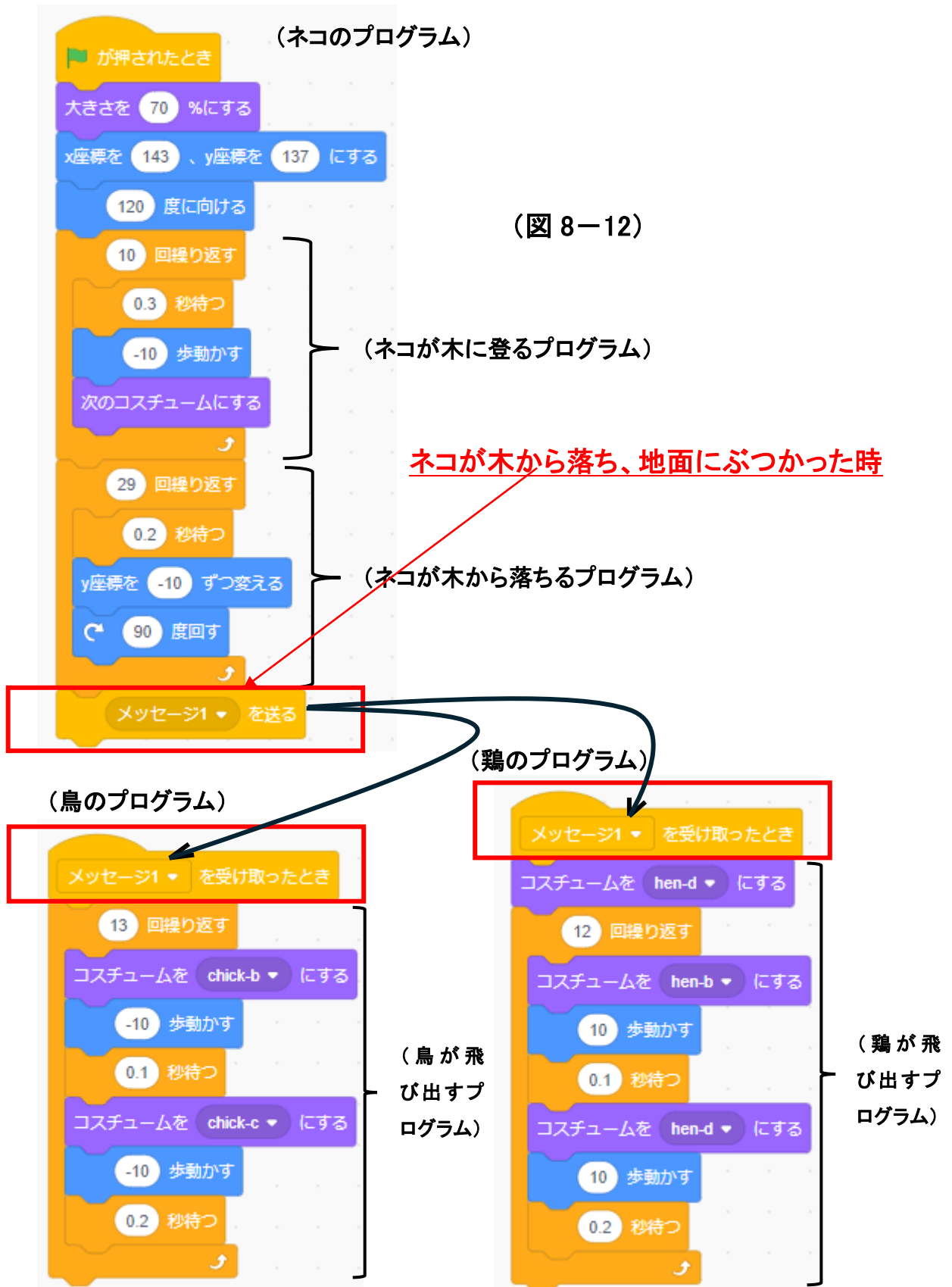
(小鳥のコスチューム)

(鶏のコスチューム)



(図 8-11)

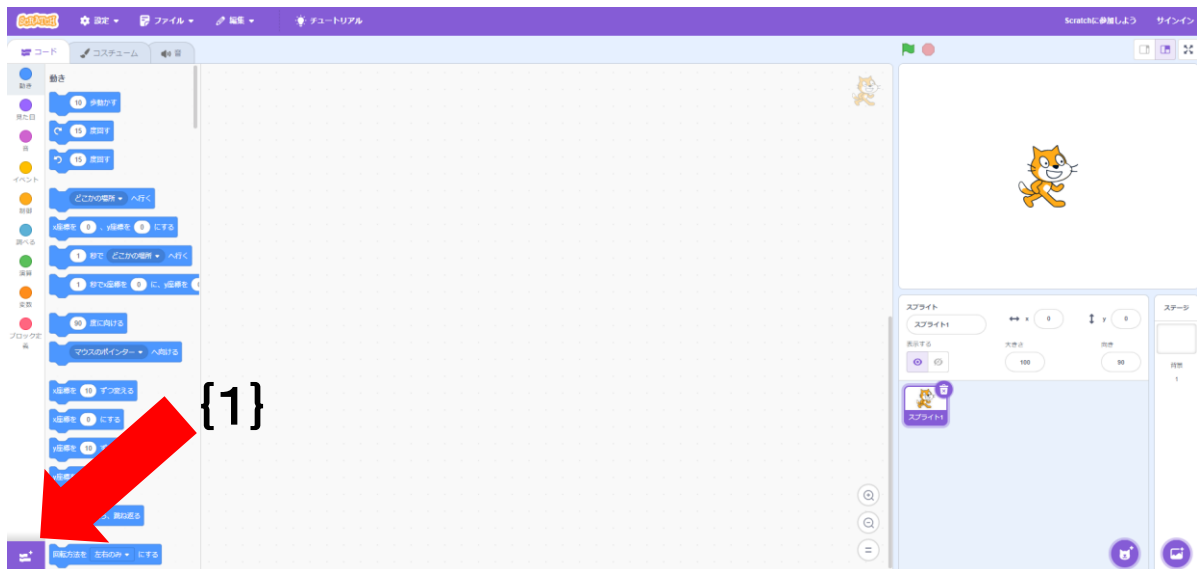
ネコと小鳥と鶏のプログラムの説明



(図 8-12)

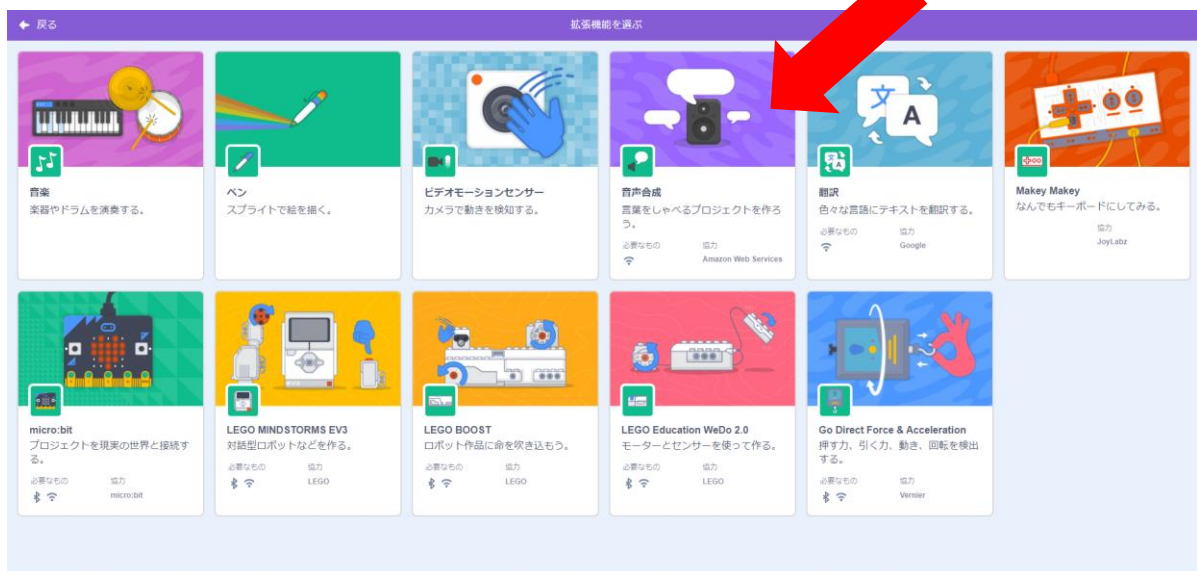
(8-5) 音声合成を使い、言葉をしゃべらせてみよう。

(1) 拡張機能をクリック。



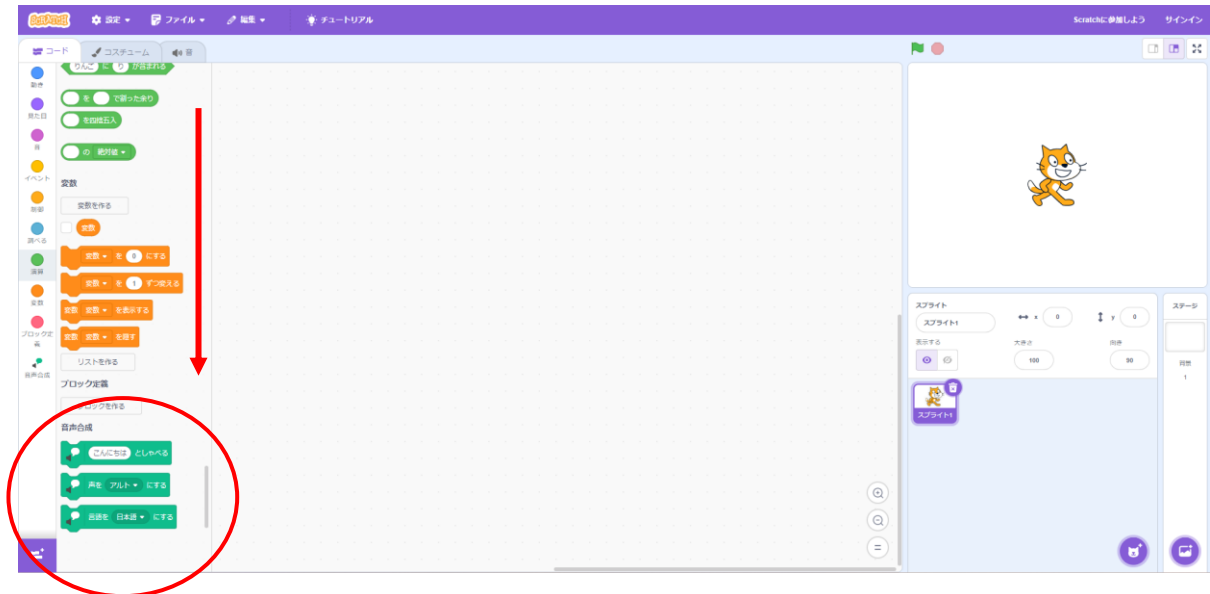
(2) 拡張機能リスト 「音声合成」をクリック

{2}



(3) 音声合成の命令の追加

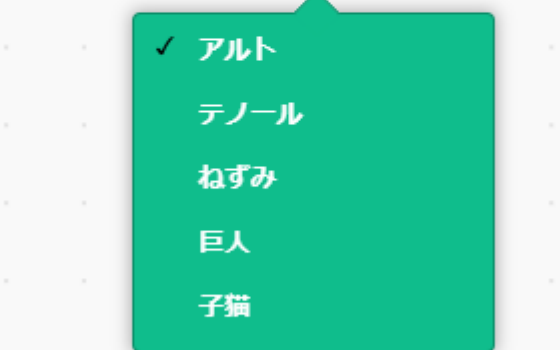
音声合成で使用出来る、3種の命令が表示されます。



この命令に「しゃべらせる言葉」を入力します。



アルト ... 女性の声
テノール ... 男性の声

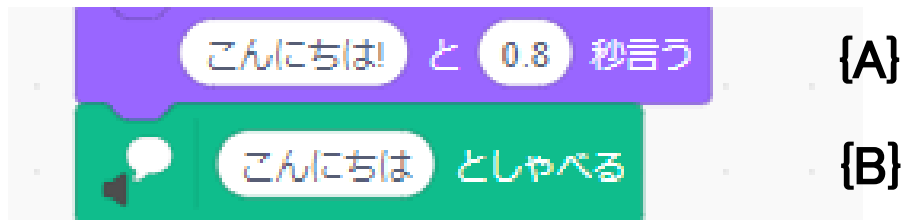


世界の言葉を話せますが、日本語ですよね。

(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重に命令を何故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、誰が話をしてるか分からない為、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。

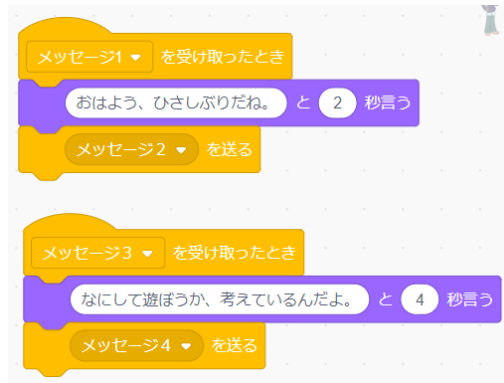
(8-6) 今日習った、「お話をする。」プログラムを使って自分のプログラムを作ってみましょう。

(女の子)



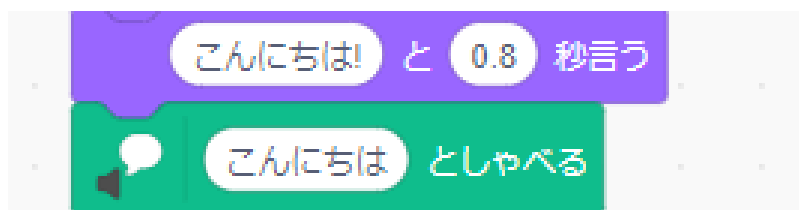
(図 8-13)

(男の子)



(図 8-14)

(音声合成)



9 プログラミング コンテストへの応募。

プログラミング教育で重要な事は、実際にプログラムを作ってみることだと思っています。よく重要な事は基礎を学ぶことだと言われてはいますが、プログラミングでは、基礎を十分に学んだとしてもプログラムを作れるかというと、そう簡単ではありません。実際にプログラムを作り始めると、思う通りに動くことはほとんどありません。その後バグを探し修正するという論理的思考を繰り返すことにより、プログラムを作り上げていきます。しかしこの過程は繰り返している内に、段々とコツを覚えていき、次第にスムーズにプログラムを作っていけるものですが。

プログラミング学習では、この論理的思考とともに、創造性の発揮をもう一つの目的としています。2・3年生の初級・中級と異なり、4年生の上級からは、どのような作品を作るか自ら考えることとなります。児童にとってこれは楽しみでもあり、又考える苦しみの始まりでもあります。しかしこの創造性こそは、社会に出ても必要な能力であり、児童に是非習得し続けて貰いたいと考えています

プログラミング学習で、素晴らしい作品を作って貰う為の方法として、次のような方法があります。まず何としても、児童にやる気を出してプログラミング学習に取り組んで貰う必要があります。その為に、プログラミング学習の最後の時間はクラス内で発表会を開くようにしています。これにより児童は、発表会に出す作品をまとめる為に、プログラム作成に更に集中する事となります。又発表を単にクラス内の発表に留まらず、プログラミングコンテスト等への応募も有効な手段と考えています。又適当なコンテストがない場合は、学校全体でのプログラミングコンテストや同じ学年だけのコンテストでも良いと考えています。要はコンテストへの応募をきっかけにして、児童がやる気を出してプログラム制作に取り込んでもらえればと考えています。

それとともにプログラム制作には、児童により違いますが、かなりの時間がかかるものです。授業時間の中でプログラム作成をしますと、時間の制限があり、作りたかった創造性に富んだ作品を残念ながら諦める事も起きてきます。そこで、この作品制作を夏休みの宿題にして貰うことがあります。これにより、児童の中には先生方が驚くような作品を夏休み明けに作ってくることがあります。本当に児童の能力の高さには驚かされます。

(9-1) “プログラミングコンテストへの応募”の為の説明。

(1) どのような作品を作ればよいか。

プログラムを作る際には、何を作るかが大変重要です。
その為の参考に。

- ① お話をつくりましょう。
 - ・他の人へ、何を伝えたいのか考えましょう。
 - ・出来れば、社会性(環境問題・地域発展・自然保護など)のあるテーマを選ぶと評価は高い。
- ② ゲームでも OK です。
 - ・見たことのあるゲームを作っても評価は低い。
 - ・ゲームであれば、工夫をして見たことのないものを。
 - ・ゲームを他の人達に遊んでもらう為に、最初にゲームの操作方法を書きましょう。
- ③ 危険なもの、他の人を傷つけるものは絶対ダメ。

(2) 何を参考に作品を作ればよいか。

作品は、単なるコピーではなく、自分で考えたものにしよう。
困った時には、次のような情報を参考にしよう。

- ① テキスト(この指導者用テキストと姉妹編の児童用テキスト(デジタル版))には、いろいろな例が載っています。参考にしましょう。
- ② スクラッチのチュートリアル(オンラインでスクラッチを開くと、最初の画面の中央に小さく出ている、初心者用教育用画面です。)を参考にしよう。
- ③ インターネットで、「スクラッチ」や「スクラッチプログラム」と検索しますと、いろいろなプログラム例が出ています。参考にしましょう。

コメント

この「プログラミングコンテストへの応募」は、補助教材
の中に、Power Point として保存されています。

(3) どうすれば、コンテストで良い結果が得られるのか。

プログラミングコンテストは、他の児童と競い合う事になります。
どうしたらよい成績を残すことが出来るか、次の4点を挙げてみました。

① **考える！**

—どのような作品を作ればよいか。プログラムはどう作れば良いのか、
どこをどのように修正すればよいのか。
考える！ ただひたすら考えるだけです。

② **友達と相談！**

—どうしても考えに詰まった時には、友達に相談してみよう。
良いアイデアが浮かぶかも知れません。友達とともに、プログラミングの
レベルが上がれば素晴らしいですね。

③ **諦めない！**

—このようなコンテストで重要な事は諦めないこと。締め切り前まで絶対に
諦めないことです。絶対に！

④ **毎日少しずつ作る！**

—プログラムを作るコツは、集中的にプログラムを作るよりも、夏休みなど
少しずつしかし必ず毎日毎日作る事です。

(4) プログラミングコンテストへの応募のメリット。

プログラミングコンテストへ応募すると、次のようなメリットが見込まれます。

① **プログラミングが間違いなくうまくなります。**

—いくつか作品を作るよりも、一つの作品を長い期間をかけて作り上げた
方が、プログラミング能力が上がります。恐らくプログラミングコンテスト
に応募した後は、先生に負けない程、プログラミングが上手になっている
でしょう。

② **将来コンピューターを扱う仕事に就く場合、きっと有利になります。**

—世界の子供達は、この Scratch を使ったプログラミングから、コンピュー
ターの世界に入っているようです。皆さんも頑張りましょう。

③ **いろいろな人達に、褒められるかもしれません。**

—お父さん・お母さん、友達、先生、いろいろな人達に、きっと褒められま
す。頑張りましょう。

④ **良い結果とともに、賞状と賞品が貰えるかもしれません。**

(9-2) プログラミング コンテストへの応募作品を作り始めましょう。

プログラムを作る際には、何を作るかが大変重要です。
その為の参考に。

- ① お話をつくりましょう。
 - ・他の人へ、何を伝えたいのか考えましょう。
 - ・出来れば、社会性(環境問題・地域発展・自然保護など)のあるテーマを選ぶと評価が高い。
- ② ゲームでも OK です。
 - ・見たことのあるゲームを作っても評価は低い。
 - ・ゲームであれば、工夫をして見たことのないものを。
 - ・ゲームを他の人達に遊んでもらう為に、最初にゲームの操作方法を書きましょう。
- ③ 危険なもの、他の人を傷つけるものは絶対ダメ。

上 級 後

一般的には、4年生で上級を習うことにより、プログラミング学習の基礎を学び終えたこととなります。それでは、5年生や6年生はどのようにプログラミング授業を続けるのか、又その際に注意すべき事柄について説明しましょう。

上級を学び終えた例えば5年生や6年生は、プログラム作品を更に作っていくこととなります。それも出来るだけ考えこまれた、創造性豊かな作品を作ることにより、論理的思考を高め、創造性を更に豊かにすることになるでしょう。

最新の動向では、5・6年生のプログラミング授業は、次のような理由によりプログラム作品制作を夏休みの課題とすることが多いようです。

- 1 上級を学んだ児童は、ある程度プログラムを自分で作る事が出来る事。
- 2 プログラミングという新たな思考への習熟にかなり個人差がある為、学校の一律の授業から離れ各自の理解力に合わせて、プログラム作品を作り上げる事が可能となる事。
- 3 創造的な素晴らしい作品を作る為には、かなりの時間がかかります。又児童の創造的作品の取り組みにも、個人差が大きいものです。それを、限られた授業の中で実現することは不可能に近く、その為にも夏休みの課題とすることの意味があると考えています。

その為に、特に1学期にプログラミング授業で注意すべき事柄を説明しましょう。

- 1 児童はこれからどのような作品を作るか、又どのようにプログラムを作るのかプログラミングの知識が少ない中で、いろいろと考え始めます。その際参考になるのが、これまで学んできたテキストです。今まで使用したプログラミングテキスト(デジタル版)を、夏休み中に自宅等でも使用出来るように準備しましょう。
- 2 夏休み後にプログラミングコンテスト等への応募を考えている場合には、そのプログラミングコンテスト等の説明を1学期中に行いましょう。
- 3 上級の授業を終えた5年生や6年生の為に、参考となるプログラミング例を書いた簡単な Power Point「Scratch 作品作成の為に」を、次ページ以降に掲載しました。どうぞご活用下さい。

5・6年生のプログラミング授業の一例。

- 1時間目 夏休み前に実施し、夏休みの課題やコンテストについて説明。
「Scratch 作品作成の為に」等を使用しプログラム作成について説明。
- 2時間目 夏休み後に実施し、クラス内での作品発表会等を開く。

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。

この作品で使用し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、この作品で使用した、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。

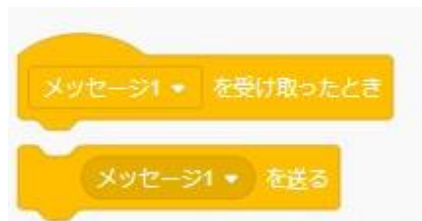


この作品では、このボタンを押す(クリック)すると、作品が始まります。

(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。

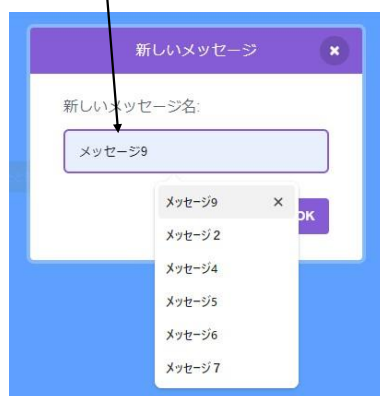


「新しいメッセージの作り方」

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。



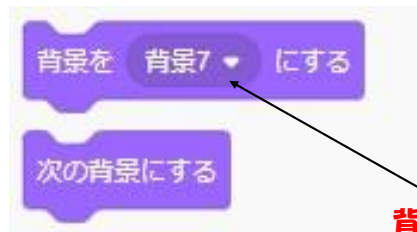
「OK」をクリックします。



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

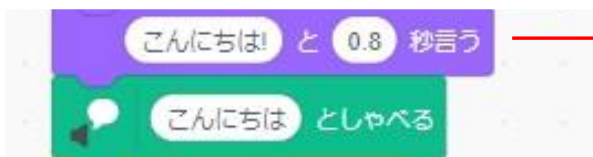
(3) 「背景」をクリック

(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。

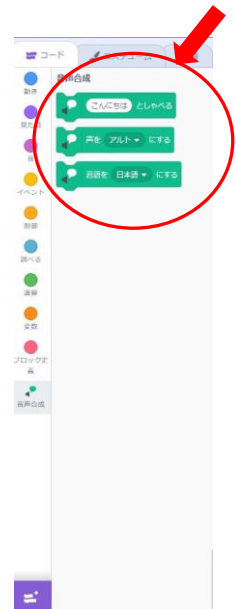
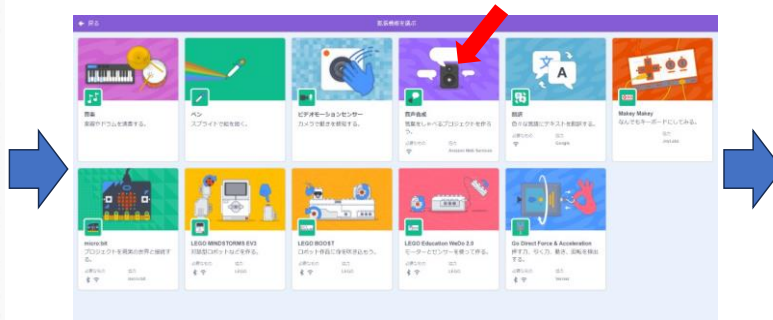
言葉を表示する命令です。
(0.8秒は「こんにちは」としゃべる大体の時間です。)



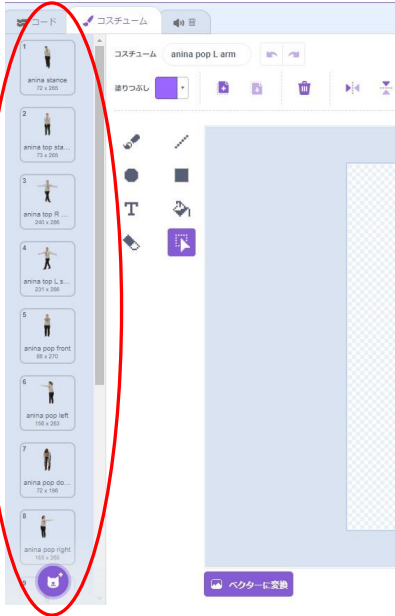
「音声合成」でしゃべる命令です。



「音声合成」の設定



5 登場人物のコスチュームの指定



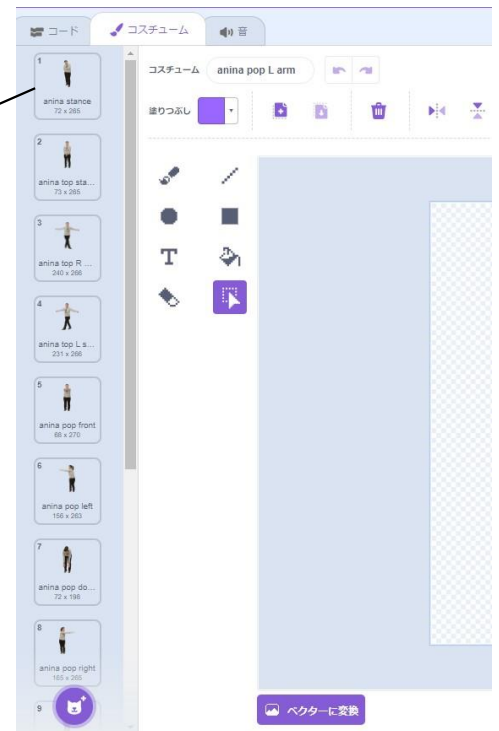
登場人物によっては、左のように多くのコスチュームを持っている場合があります。

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！



『Scratch は、MIT メディア・ラボのライフロング・キンダーガーデン・グループの協力により、Scratch 財団が進めているプロジェクトです。
<https://scratch.mit.edu> から自由に入手できます。』

初版	2017年10月	児童用テキスト初版
2版	2018年 8月	児童用テキスト。一部修正追記。
3版	2020年 2月	児童用テキスト。修正追記。
4版	2021年11月	児童用テキスト。応用編を追加。一部修正追記。
5版	2022年 9月	児童用テキスト。Digital text。応用編を追加。一部修正追記。
6版	2023年 4月	児童用テキスト。導入編を追加。一部修正追記。
7版	2024年12月	指導者用テキスト。全面更新。

おわり