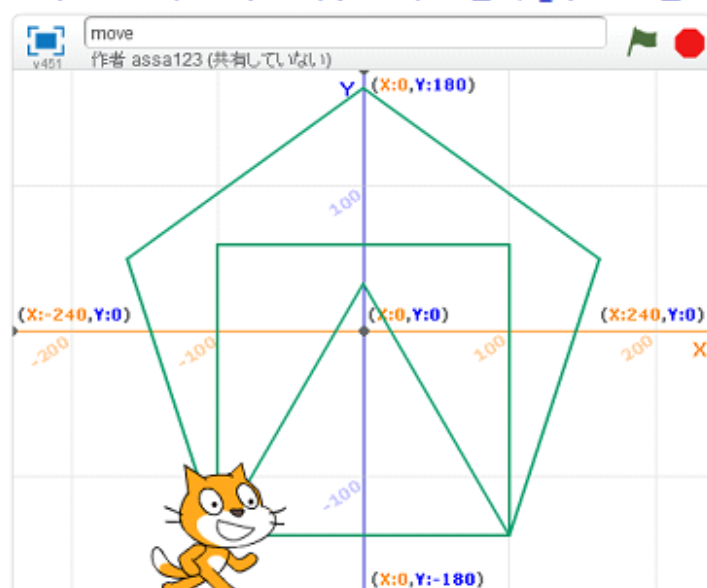


Scratchで プログラミングを楽しもう



河西 朝雄 著



KASAI, SOFTWARELAB

Scratch ってなんだろう

■Scratch の特徴

Scratch はマサチューセッツ工科大学メディアラボで開発された、子ども向けの**プログラミング**環境です。

一般の**プログラミング言語**では、多くの命令を覚えて、コードを打ち込み、プログラムを書きます。一方、Scratch ではスクリプトというブロックをつなぎ合わせることでプログラムを書きます。コードを打ち込む必要がないため、文法の誤りを気にせずにプログラミングできます。

※「プログラミングとは」「プログラミング言語とは」

皆さんはゲーム機でゲームをすると思います。ゲームの動きをゲーム機（コンピュータ）に指示するものがプログラムで、プログラムを作ることをプログラミングといいます。人間は日本語や英語で話をしますが、人間がコンピュータに話をする（指示する）言葉をプログラミング言語といいます。

また操作方法も基本的にマウスによる**ドラッグ&ドロップ**のみであるため、比較的簡単にプログラミングできる仕様となっています。

従って Scratch はプログラミング初心者やコンピュータに不慣れな者にも、簡単にプログラミングに入門できる環境です。

■教育現場でも注目が集まっています

プログラミングは、中学校の「技術・家庭」や高校の教科「情報」で教えられています。また、小学校の授業にも取り入れる動きが出てきました。アメリカをはじめ世界各国でもプログラミング教育の機運が高まっており、日本もその流れに乗ろうとしています。

小中学生にプログラミングの基礎を教えるには Scratch は最適です。高校生にも十分対応できます。IT 企業も「グリー×教育プロジェクト」などで Scratch プログラミングを推奨しています。次期学習指導要領が始まる 2020 年度からコンピューターのプログラミング教育を小中学校で必修にする計画です。

■2021 年度からの大学入試に対応

現在の中学 1 年生が大学受験をする 2021 年度から大学入試制度が大きく変わります。習得した「知識・技能」の確認に留まらず、「知識・技能を活用する力」、つまり総合的な「思考力・表現力」を問う検査が重要になります。こうした思考力・表現力を身につけるにはプログラミングは最適な教材と言えます。

■このテキストについて

・このテキストは小学校高学年（主に5年生、6年生）、中学生、高校生を対象にしています。

・第1回～第10回までが以下のような構成になっています。各回を45分～50分の授業で学習します。

- 第1回 Scratch でねこを動かしてみよう
- 第2回 アニメーションに挑戦そのⅠ
- 第3回 アニメーションに挑戦そのⅡ
- 第4回 図形を描いてみようそのⅠ
- 第5回 図形を描いてみようそのⅡ
- 第6回 プログラムのデータを管理する変数
- 第7回 データをまとめて管理するリスト
- 第8回 プログラムを制御するそのⅠ
- 第9回 プログラムを制御するそのⅡ
- 第10回 同じ処理をひとつにまとめるブロック

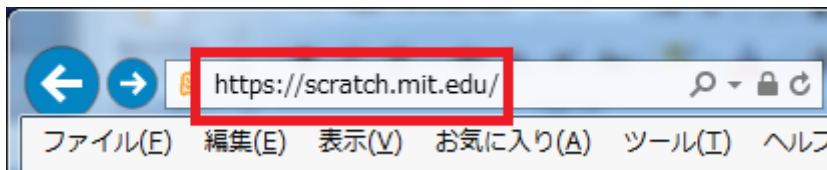
第1回 Scratch でねこを動かしてみよう

1. Scratch を始めるには？

■Scratch の Web ページからユーザー登録する

- Web ページアドレス (URL) を入力。

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



- 「Scratch に参加しよう」を選択。



- 登録するための質問に答える。「次へ」で次の質問に進む。

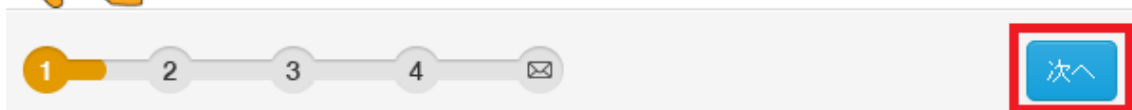
Scratchに参加しよう

Scratchのアカウント作成はとても簡単！（そして無料です）

Scratchで使うユーザー名を入力

パスワードを入力

パスワードの確認



■Scratch にサインイン

- ・ユーザー名とパスワードを入力。



Scratch!に参加しよう サインイン

ユーザー名

This field is required

パスワード

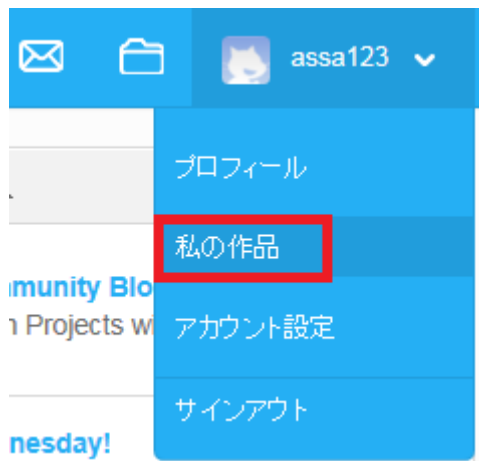
This field is required

サインイン

パスワードを忘れた場合。

■プログラムを作る

- ・「私の作品」を選択。

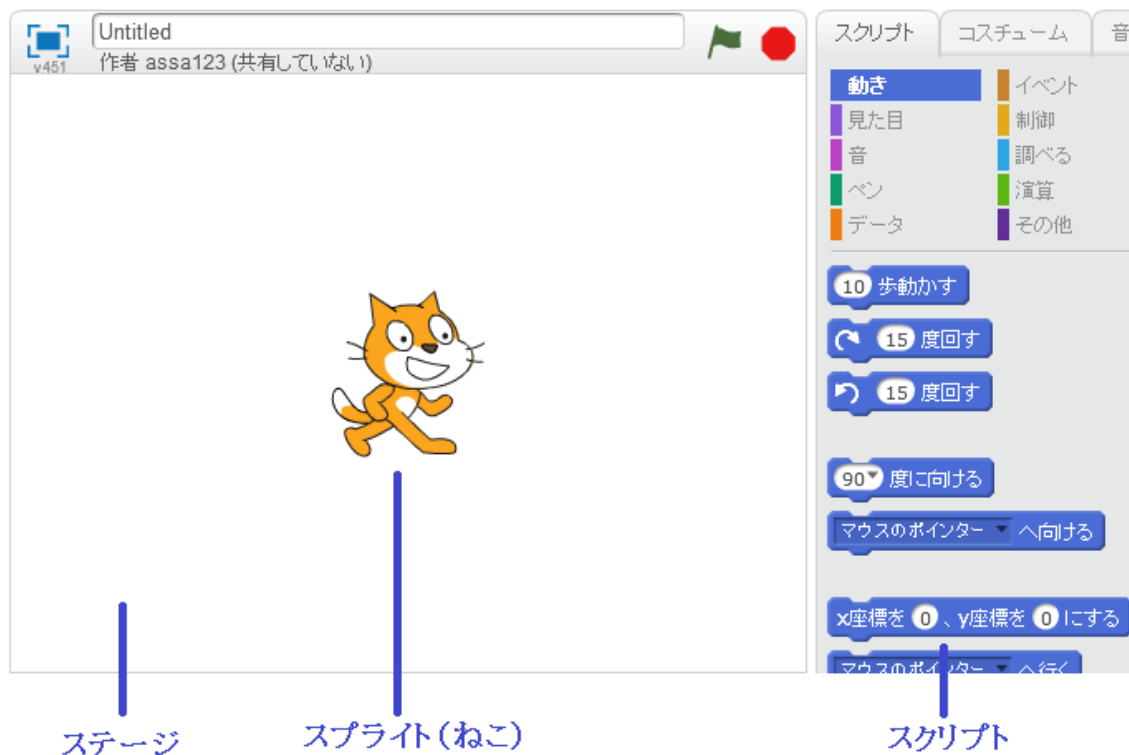


- ・「新しいプロジェクト」を選択。

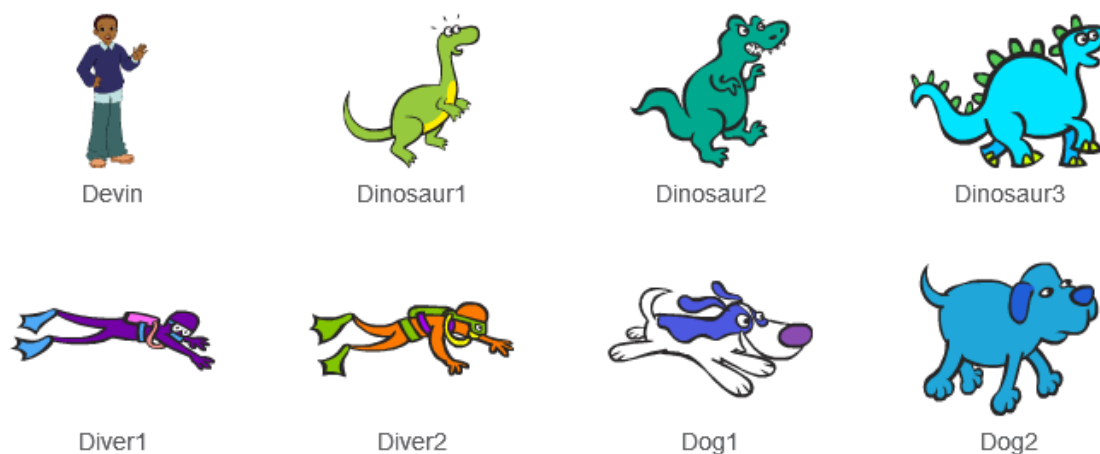


2. Scratch の画面

- Scratch でプログラムを作る時の主役は「ねこ」です。
- 最初、ステージの真ん中にいます。ステージは「ねこ」が動きまわる場所です。
- このステージの中の「ねこ」をスクリプトを使って動かします。スクリプトはプログラムするときの命令（指令）です。



※「ねこ」以外のキャラクターも使えます。Scratch でプログラムするとき画面で動くキャラクターを総称してスプライトといいます。




3. ねこを動かしてみよう

「スクリプト」と書かれた枠の中に青色の**スクリプト**がいくつかあります。

※「スクリプトとは」Scratch でプログラムするときの命令（指令）のことです。



①  をマウスでクリックしてください。

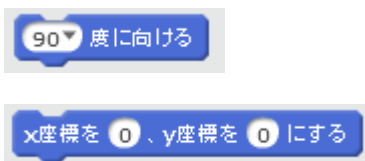
ねこが 10 歩右に移動します。ここでいう 10 歩というのは 10 **ドット(ピクセル)**のことです。

※「ドット（ピクセル）とは」画面に映る画像は細かな点（赤、緑、青）でできています。この点のことをドットとかピクセルといいます。

②  をマウスでクリックしてください。

ねこが 15 度だけ時計回りに回転します。

③ねこを元の位置と向きに戻すには以下の 2 つのスクリプトをクリックしてください。



4. プログラムを作ってみよう

・「スクリプト」にあるスクリプトをマウスでクリックすると単独の動きはできますが、一連の動作（プログラム）はできません。

■スクリプトを並べてプログラムを作ります

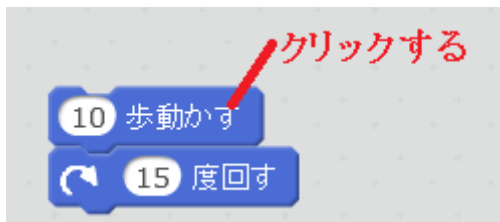
・「スクリプト」にあるスクリプトを画面右のスクリプトエリアにドラッグドロップします。

以下の例では「10 歩動かす」と「15 度回す」をドラッグドロップしました。2つのスクリプトを近づけて置くと、スクリプトの上下にある凸凹が連結されます。



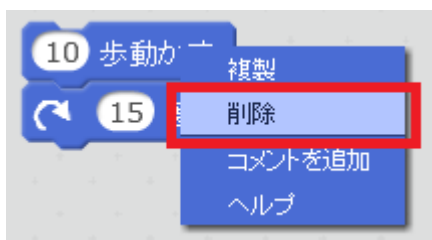
■プログラムの開始

・スクリプトエリアにドラッグドロップしたスクリプト（どこでも良い）をクリックすると、プログラムが開始されます。



■ スクリプトの削除

- ・ 右クリックして「削除」を選択します。



- ・ またはスクリプトブロックにドラッグドロップします。

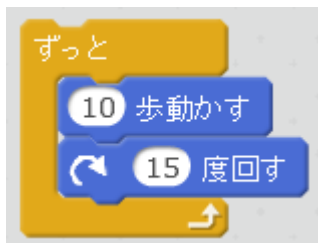


5. 何度もくりかえすには？

- ・「スクリプト」の「制御」を選択します。黄色のスクリプトです。
- ・「ずっと」というスクリプトをドラッグドロップします

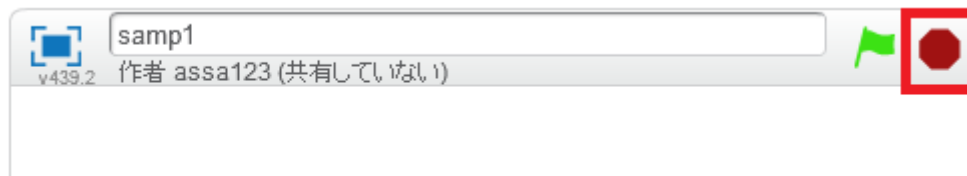


- ・「ずっと」の中に繰り返したいスクリプトを入れます



■プログラムの停止

- ・プログラムを停止するには「赤色の八角形」をクリックします




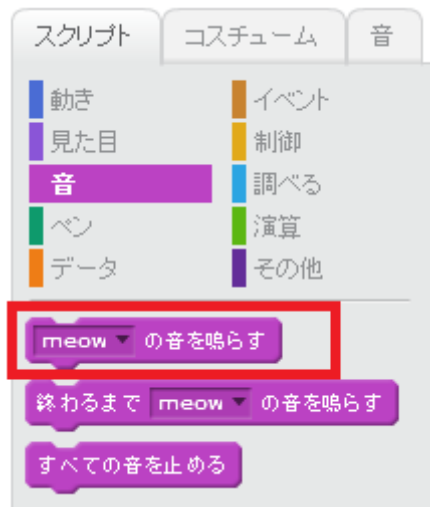
6. 何かしゃべってみよう

- ・ねこは動くだけでなく、しゃべることもできます。

■「音」スクリプト

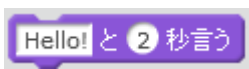
- ・「スクリプト」の「音」を選択します。赤紫色のスクリプトです。

 をスクリプトエリアにドラッグドロップします。このスクリプトは「みゃー」という鳴き声を出します。



■「見た目」スクリプト

- ・「スクリプト」の「見た目」を選択します。紫色のスクリプトです。

 をスクリプトエリアにドラッグドロップします。このスクリプトは「Hello!」というメッセージを2秒間吹き出しで表示します。



■ スクリプトの内容を変えることができます

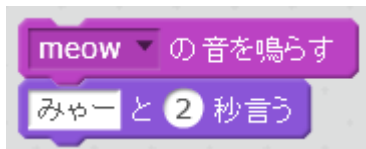
「Hello!」位置をマウスでクリックしマウスカーソルを移し、



内容を「みゃー」に打ち直します。



・ 以上から完成したプログラムです。



7. 大きさを変えてみよう

■ スプライトの大きさを変えるスクリプト

「スクリプト」の「見た目」を選択します。紫色のスクリプトです。

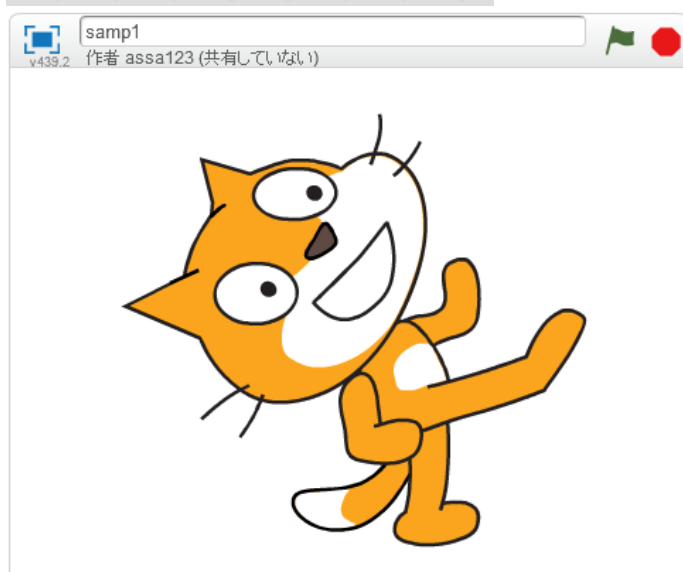
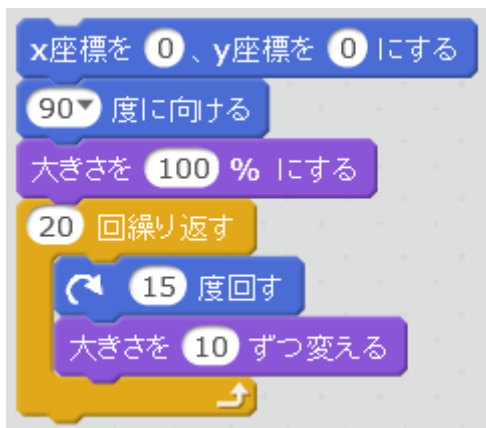
- ・スプライトの大きさを指定した%で変えます。

大きさを **10** ずつ変える

- ・スプライトの大きさを指定した%に設定します。100%が原寸、200%は倍のサイズとなります。

大きさを **100** % にする

- ・15度回転しながら大きさを10%ずつ増やす動作を20回繰り返すプログラムです。



8. スタンプ機能で「ねこ」のコピーを作ろう

- ・「スクリプト」の「ペン」を選択します。緑色のスクリプトです。



- ・ステージに書かれた内容を消します。

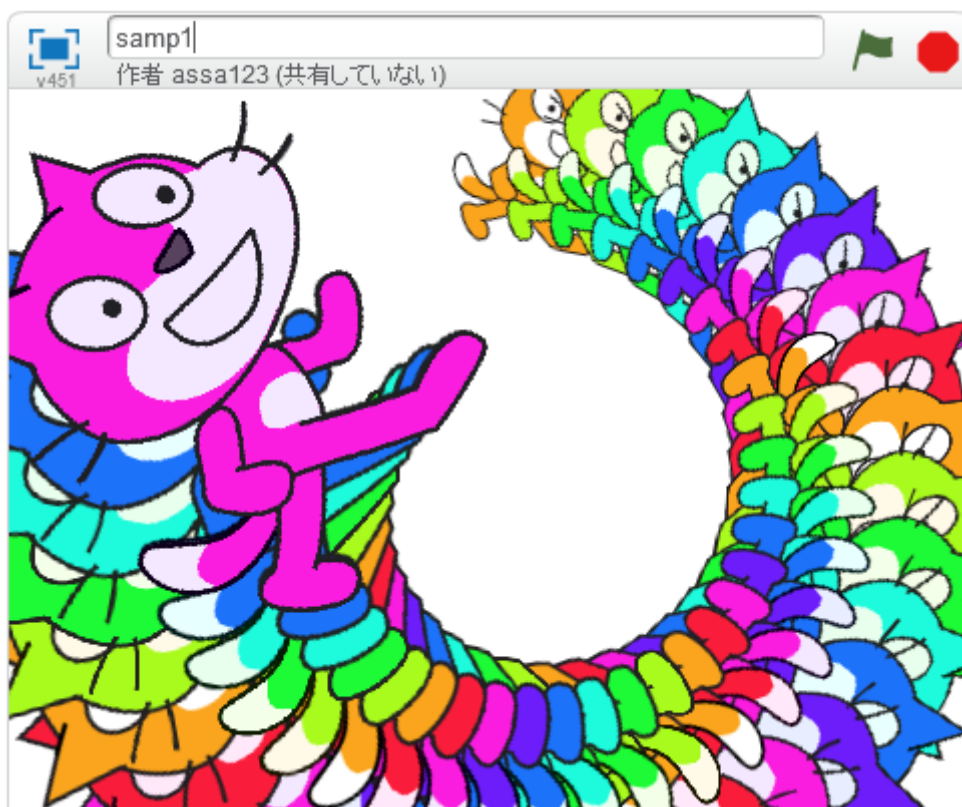


- ・スプライトのコピーをステージに描きます。



- ・スタンプ機能を使って「ねこ」のコピーをステージに描きます。
「ねこ」を回転しながら、大きさと色を変えます。






第2回 アニメーションに挑戦その I

1. ねこの代わりに恐竜を使ってみよう

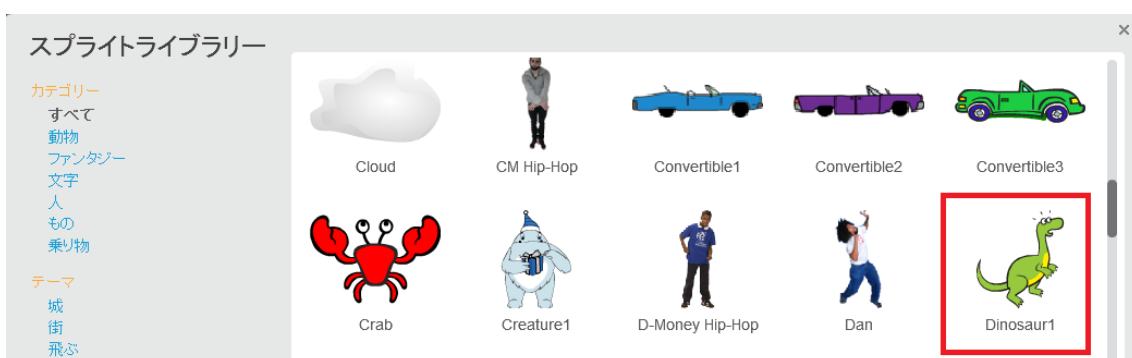
- ・ねこのスプライトを右クリックして「削除」を選択します。



- ・「新しいスプライト」の  を選択し、スプライトをライブラリから選択します。

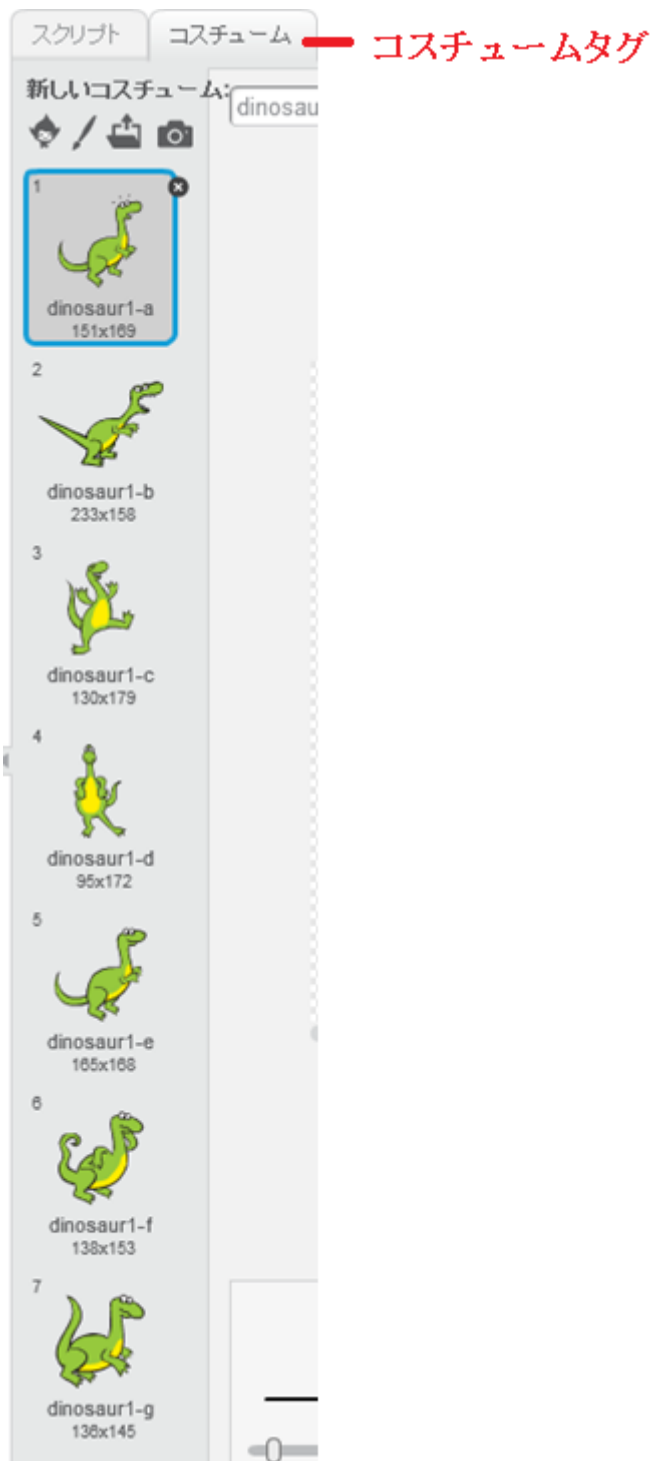


- ・希望するスプライトを選択し「OK」ボタンをクリックします。



2. コスチュームを変えてみよう

・コスチュームはスプライトの色々な容姿です。どのようなコスチュームがあるか調べるにはコスチュームタグを選択します。恐竜のスプライトは7種類のコスチュームを持っています。

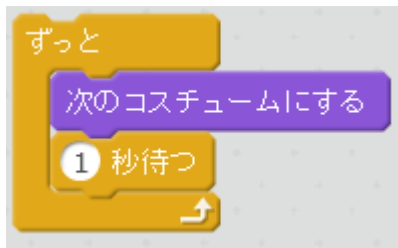


- ・プログラムで次のコスチュームに変えるには「スクリプト」の「見た目」の

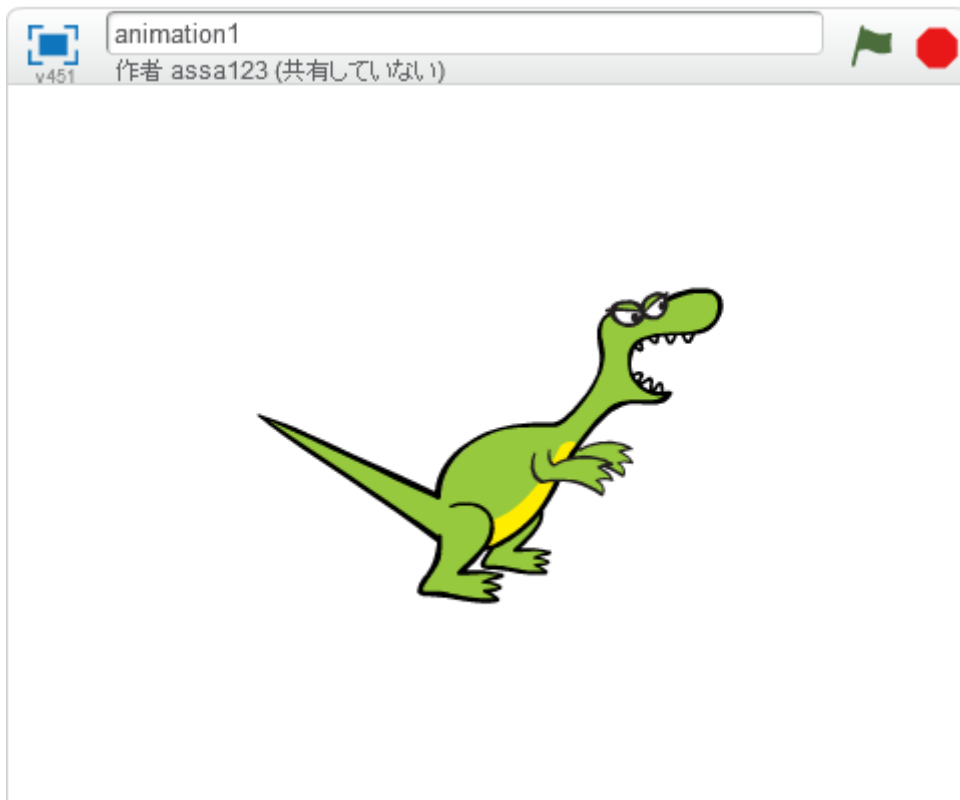
次のコスチュームにする
を使います。



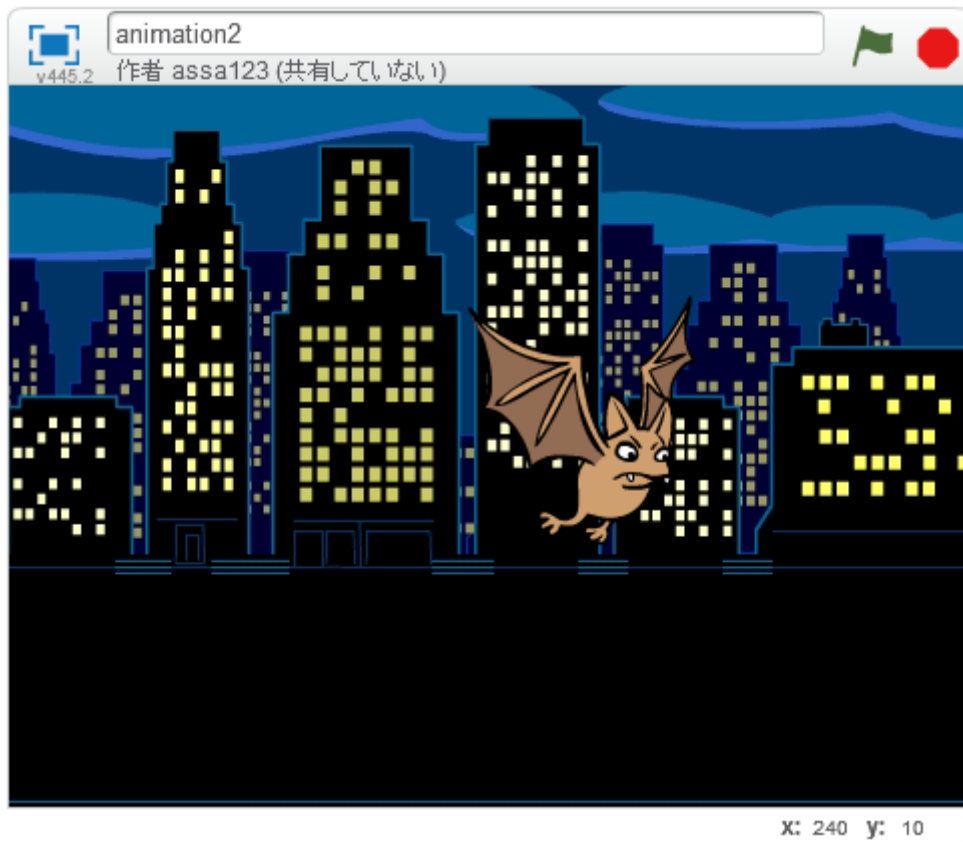
■恐竜のコスチュームを1秒ごとに変えるプログラムです。



・コスチュームの切り替えに少し時間をおくには「スクリプト」の「制御」の「1秒待つ」を使います。



3. こうもりを飛ばそう

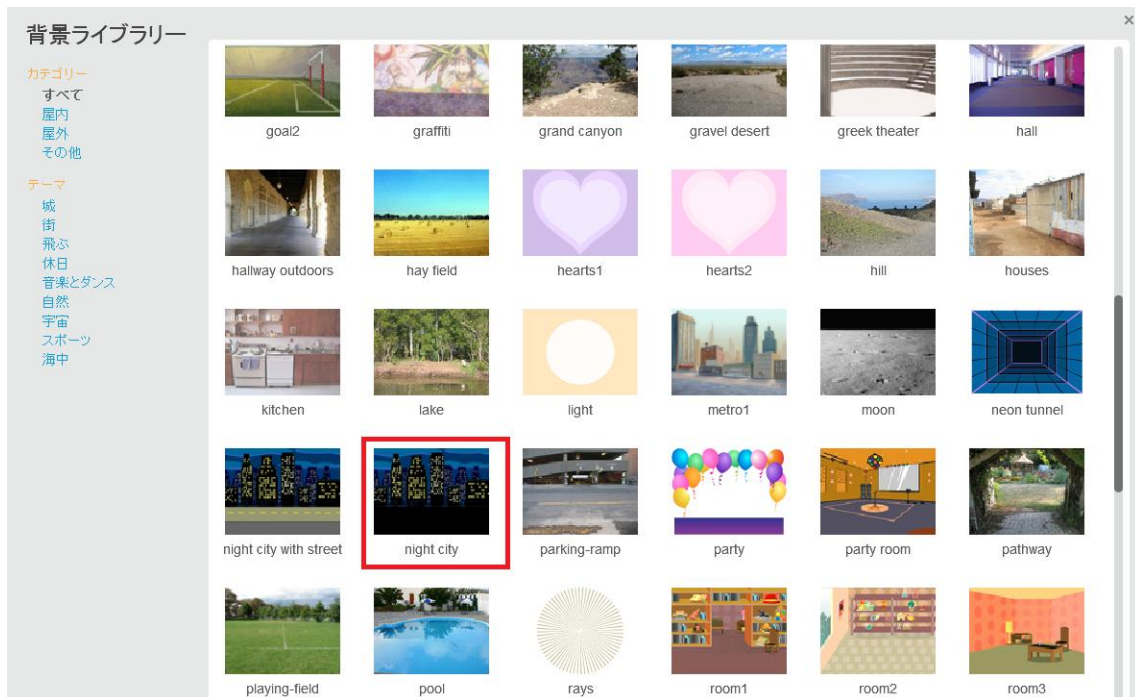


■背景を変える

- ・新しい背景を選択



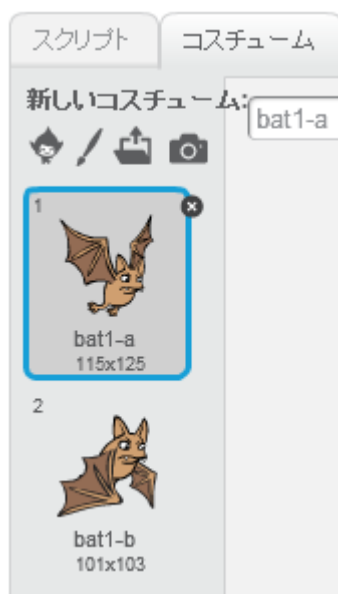
- ・「背景ライブラリー」から night city を選択



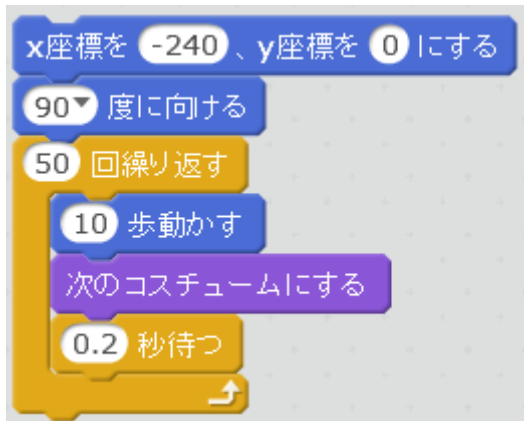
- スプライトをこうもり (Bat1) に変える



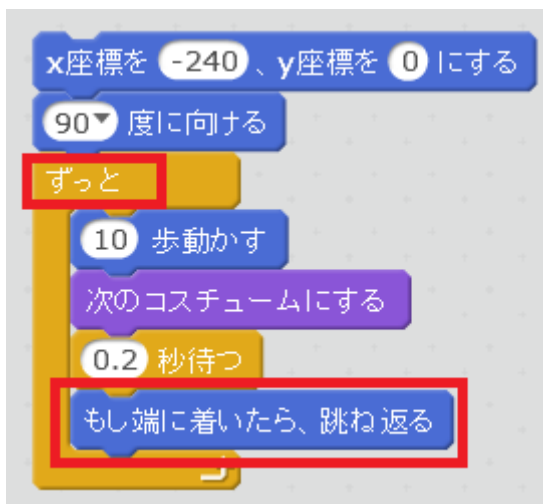
- ・ Bat1 のコスチュームは 2 種類



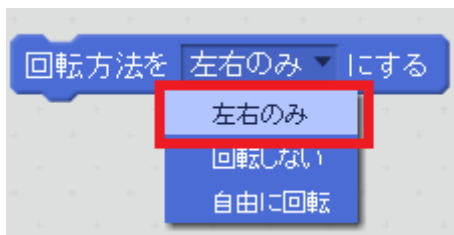
■ こうもりを画面の左から右へコスチュームを変えながら移動する



■ こうもりを画面の端で跳ね返す



■ 跳ね返った時にこうもりの上下が逆にならないようにする



第3回 アニメーションに挑戦そのⅡ

1. ダンスを踊ろう




■背景を spotlight-stage、スプライトを Ballerina と Catherine にする

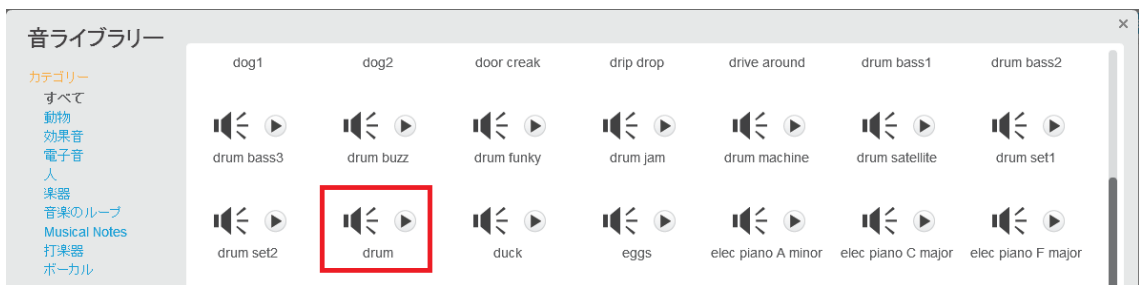


■音の登録

- ・音はスプライトごとに登録します。Ballerina や Catherine には標準で pop という音が登録されています。




- ・ を選ぶと以下のような音ライブラリが表示されますので、希望の音を登録します。
今回は Ballerina に「drum」を登録します。



■音を鳴らす

- ・「スクリプト」の「音」を選択します。紫色のスクリプトです。

- ・ で登録した drum の音を鳴らします。



■緑旗でプログラムを開始

- 2つのスプライトを同時に動かすには緑旗をクリックします。



- このイベントを受け取るには以下のスクリプトを使います。

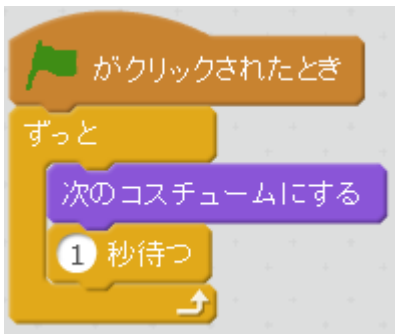


■スクリプト

- Ballerina



- Catherine



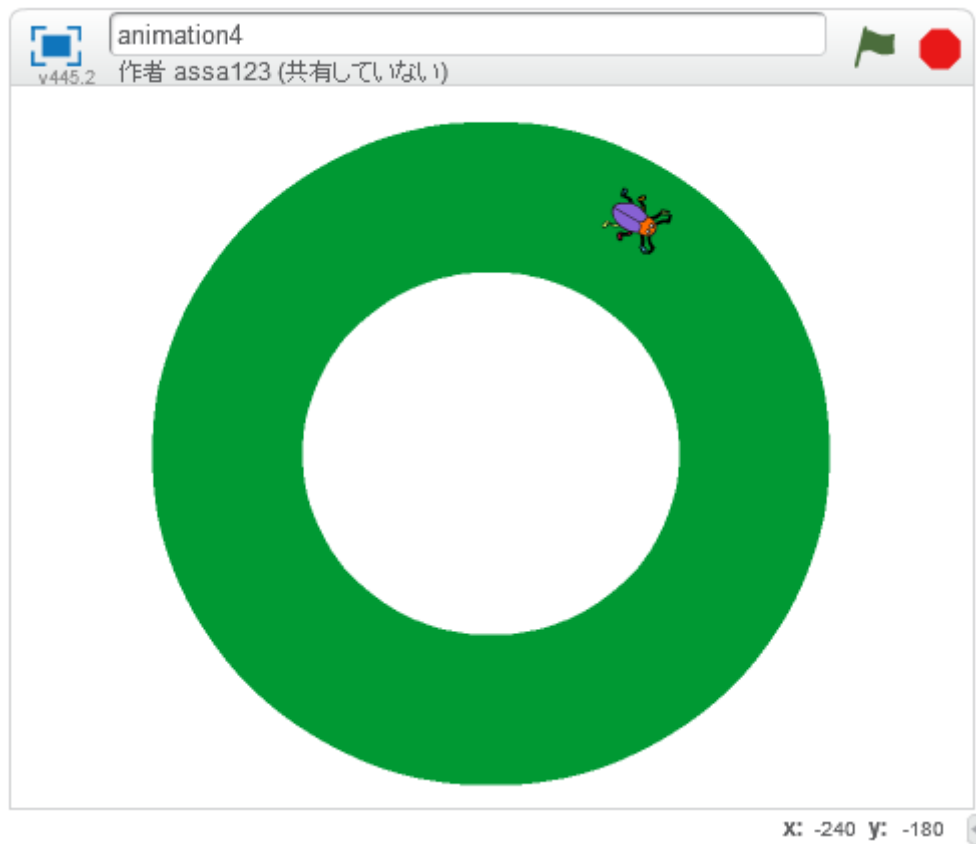
☆やってみよう

- 3人目のスプライトに「Adrian」を追加してみよう



Adrian


2. 運動会




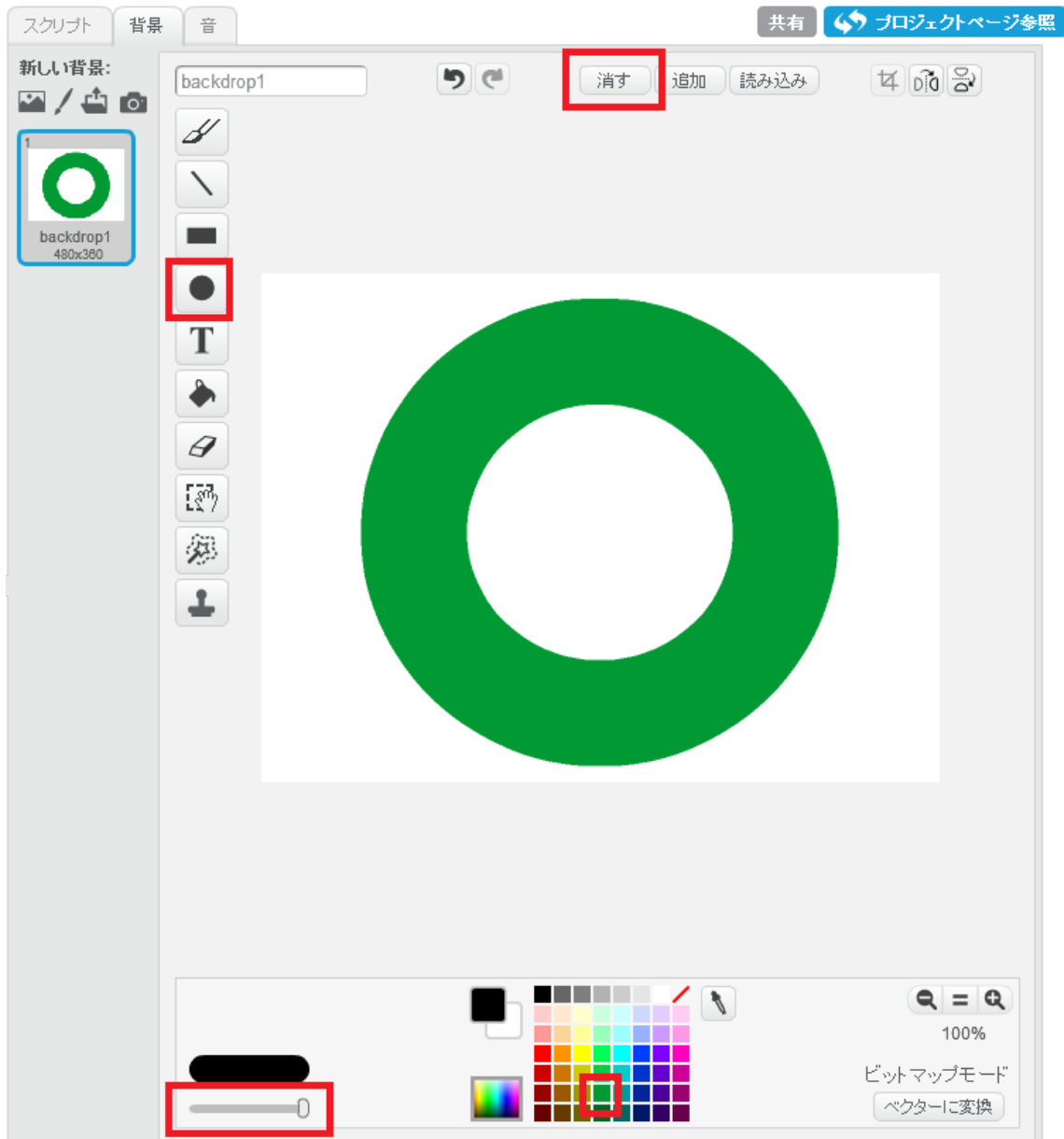
■背景を描く

- 新しい背景:
を選択

- で線の太さを選択

- で色を選ぶ

-  を選択し、背景描画画面でマウスをドラッグして円を描く



- やり直すには「消す」で消す

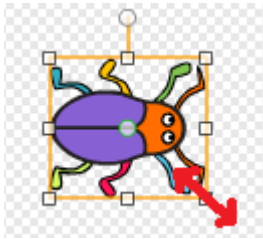
■ スプライトの縮小とセンタリング


スプライトを **Beetle** にする。Beetle は大きいので縮小します。

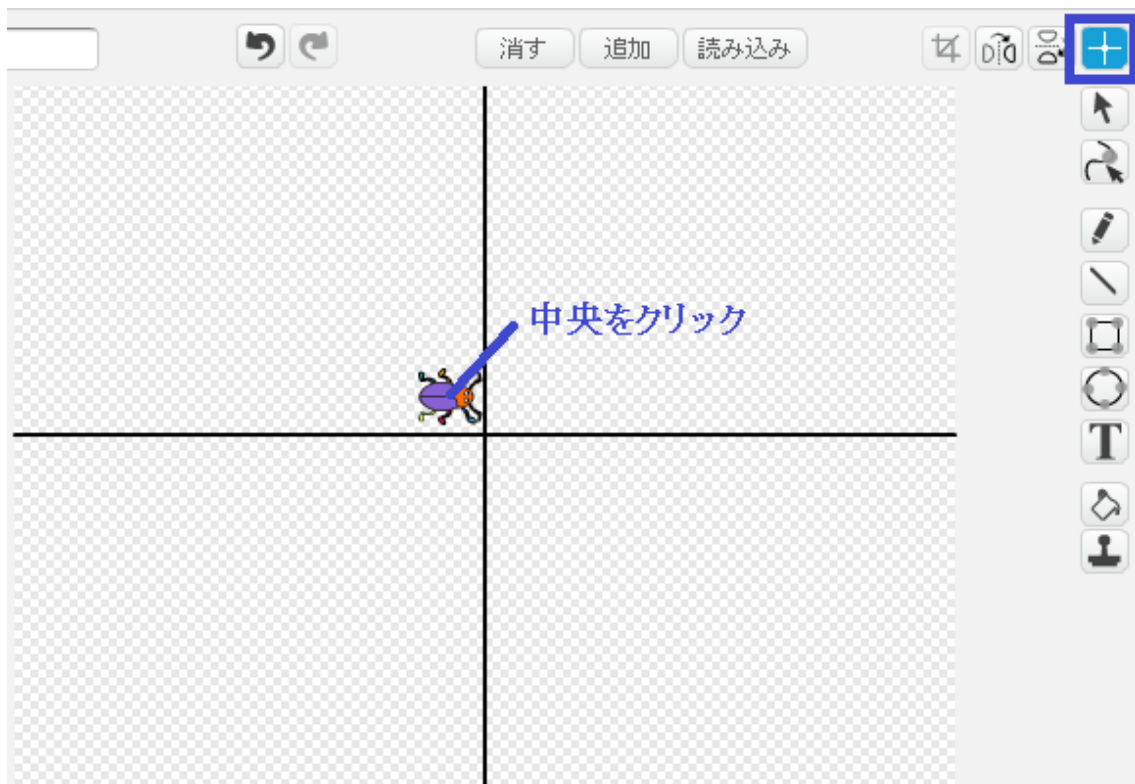
- コスチュームを選びます。



- **Beetle** を選択し、マウスをドラッグして縮小します。



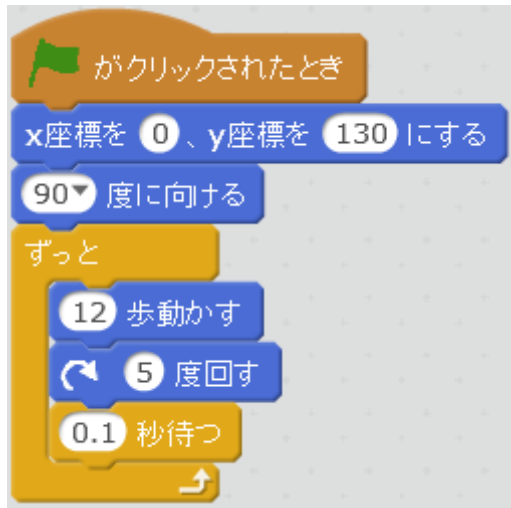
-  を選択し、**Beetle** の中央をクリックします。これで **Beetle** の中心がセンタリングされます。



■ スクリプト

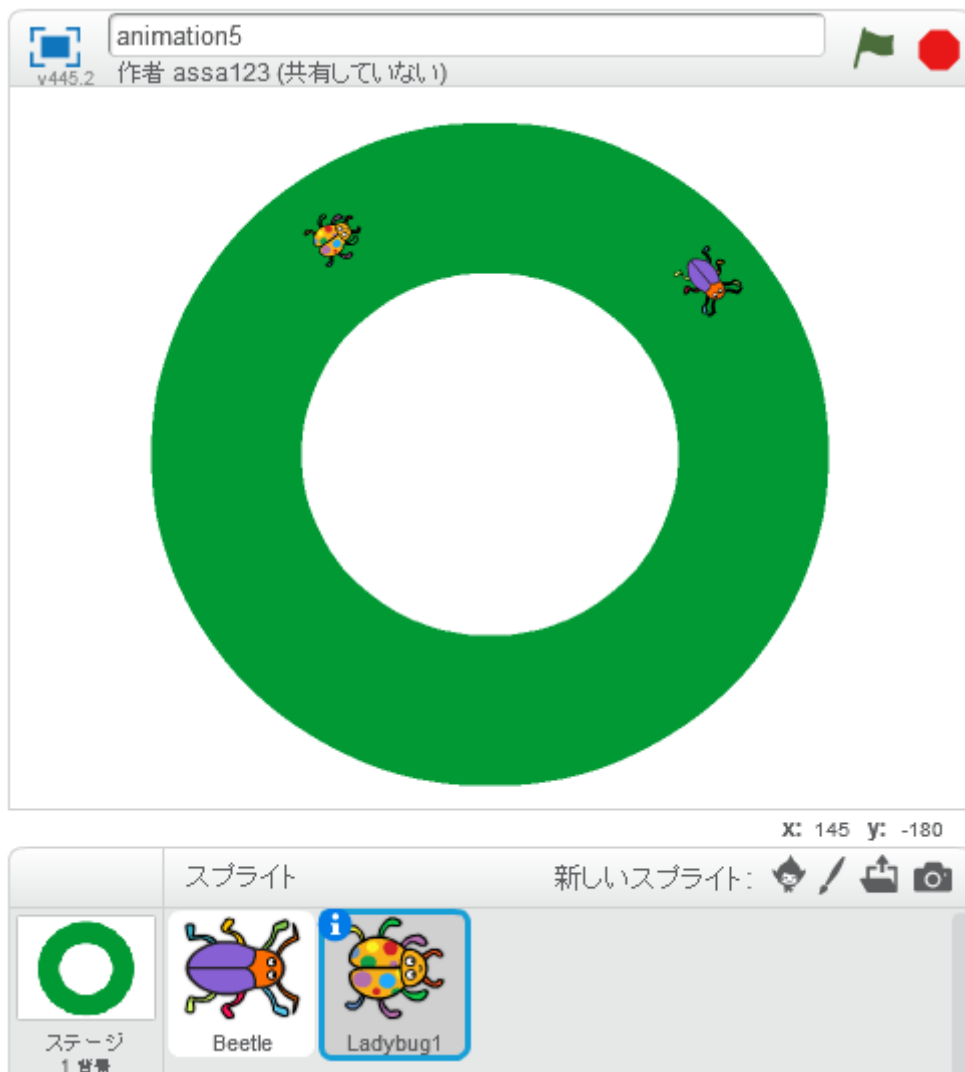
Beetle のスタート地点は (0,130) で、12 歩動かして 5 度回転することを繰り返すことで円周上を移動します。

• Beetle



☆やってみよう

- スプライトに Ladybug1 を追加して 2 匹で競争してみよう。
- Ladybug1 の方が遅く動くようにするにはどうしたら良いだろうか。



第4回 図形を描いてみようその I

1. 指定した座標にねこを移す

■位置を示す座標とは

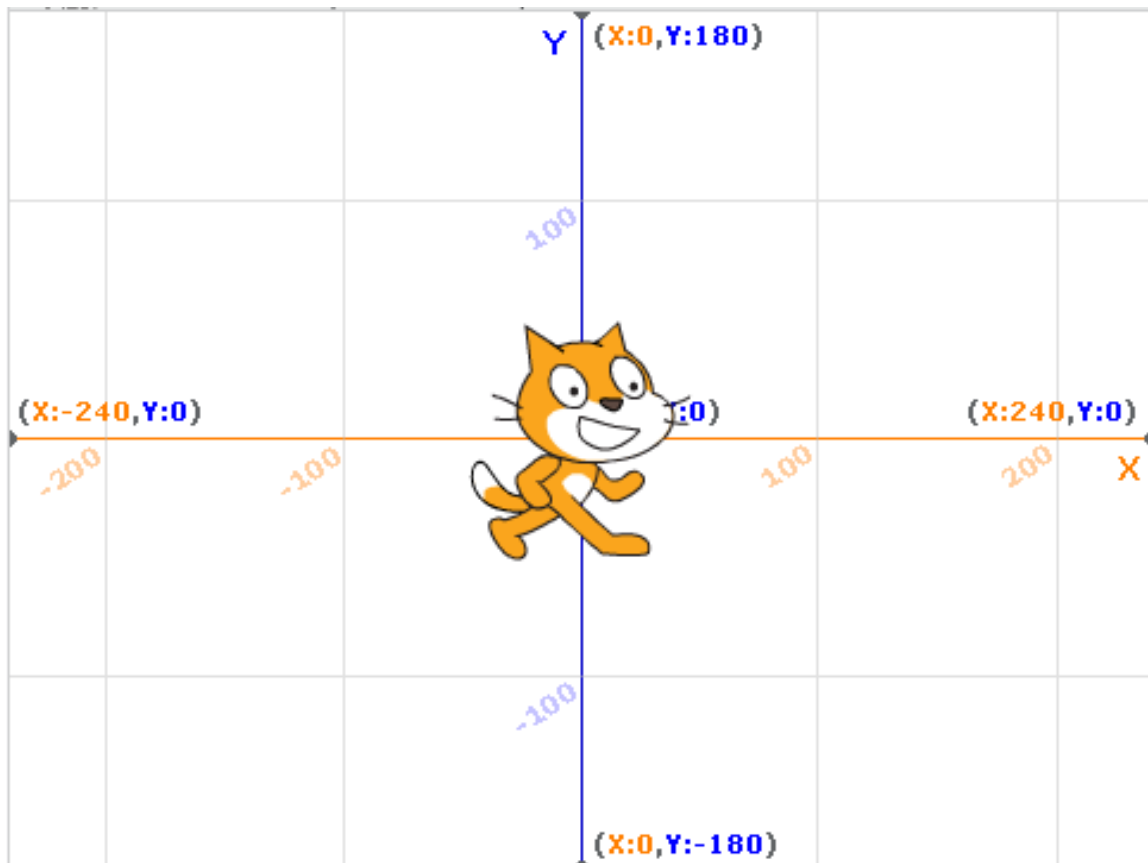
- ・ねこの位置は **x 座標**、**y 座標** で指定します。
- ・数学（算数）では位置を決めるのに座標を使います。画面の横方向を x 座標、縦方向を y 座標と呼びます。

※「X座標、Y座標とは」

Xは横軸、Yは縦軸を表しています。数字だけの場合は、比例と反比例で学習したように、X軸の場合は中心より右、Y軸の場合は中心より上を表します。数字にマイナスがついていたら、X軸の場合は中心より左、Y軸の場合は中心より下を表します。

- ・画面の中心が (0,0) 位置となります。
- ・右方向が x 座標の正の方向、上方向が y 座標の正の方向となります。
- ・Scratch で使う画面の座標は以下のようにになっています。
- ・画面の大きさは横（x 座標）が-240~240 で、縦（y 座標）が-180~180 です。

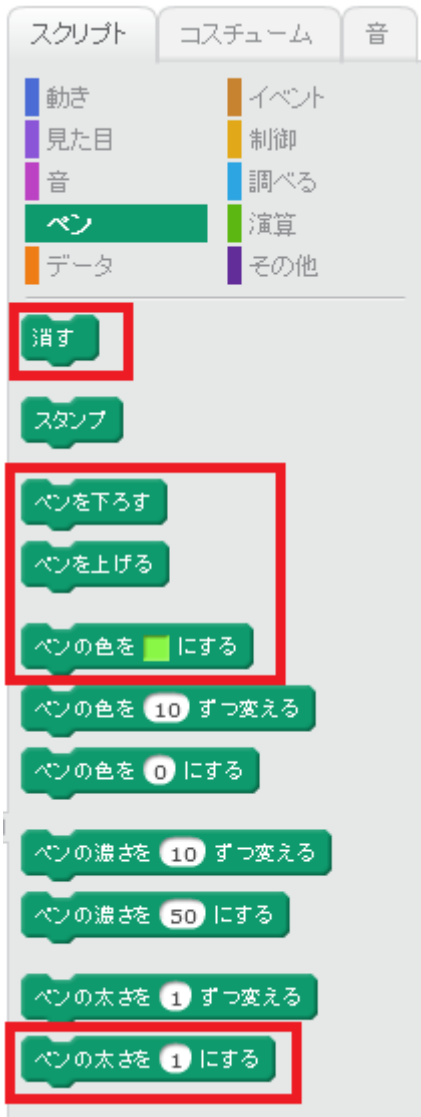
以下の背景は「xy-grid」を選択します。



■ペン機能

ステージにペンを使って描画を行うことができます。

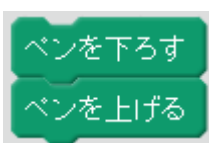
- ・「スクリプト」の「ペン」を選択します。緑色のスクリプトです。




- ・ステージに描画されているものをすべて消します。



- ・ペンの上げ、下げを指定します。



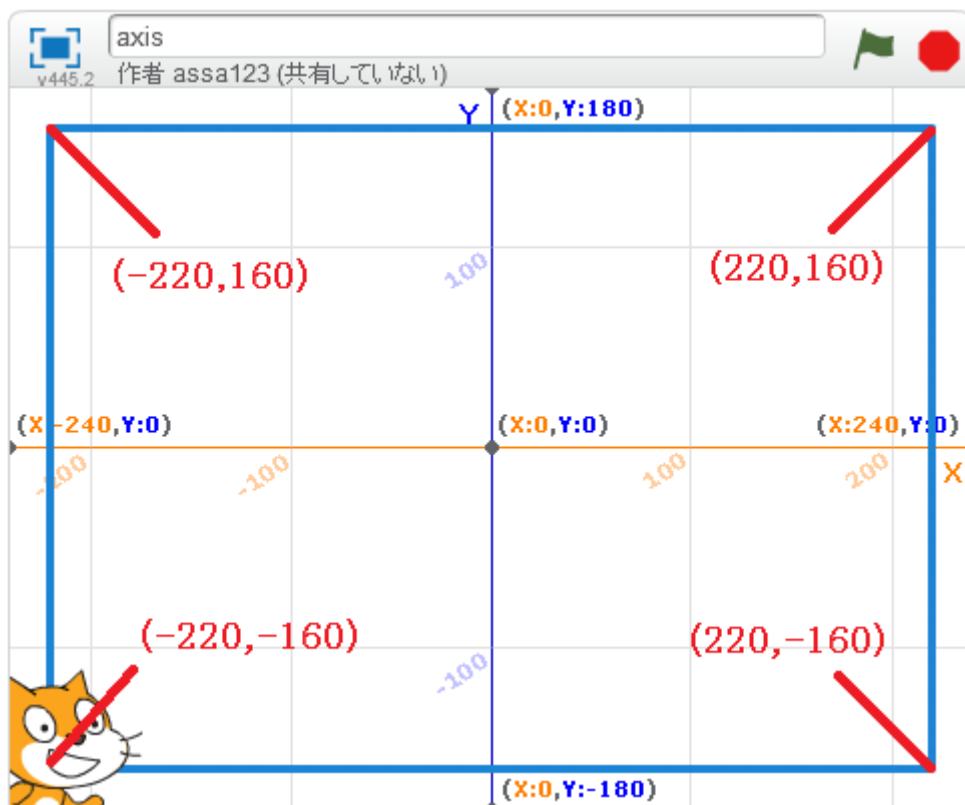
・ペンの色を指定した色に設定します。色を指定するにはをクリックするとマウスがハンドマークになるので、画面上のスプライトや、スクリプトなどの色をクリックします。するとその色が選ばれます。

ペンの色を  にする

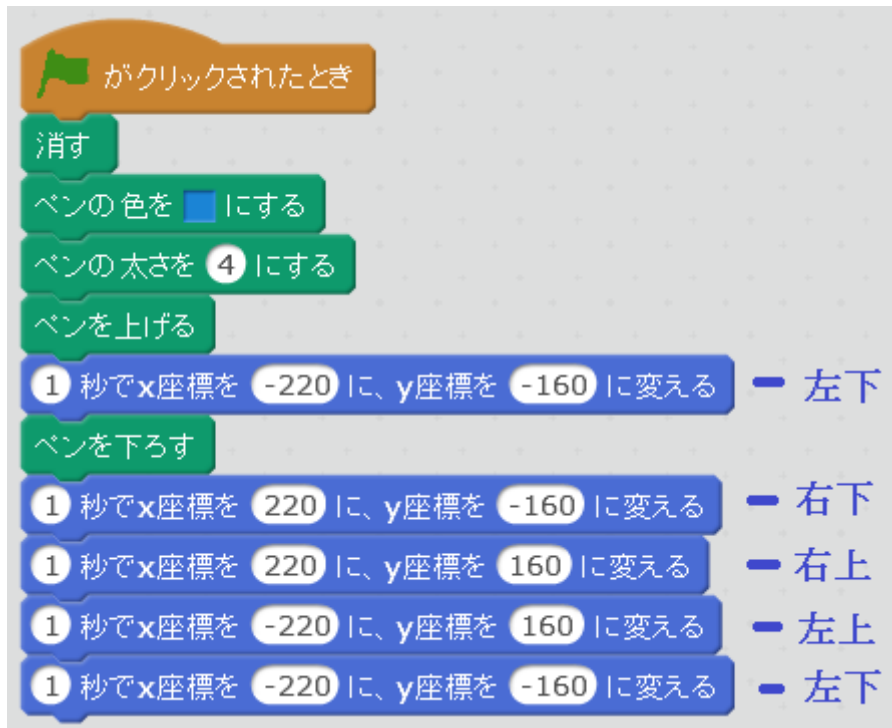
・ペンの太さを指定した数値に変えます。

ペンの太さを **1** にする

■ 4 隅を移動するには画面の
左下(-220,-160)をスタートして、
右下(220,-160)、
右上(220,160)、
左上(-220,160)、
左下(-220,-160)と移動します。

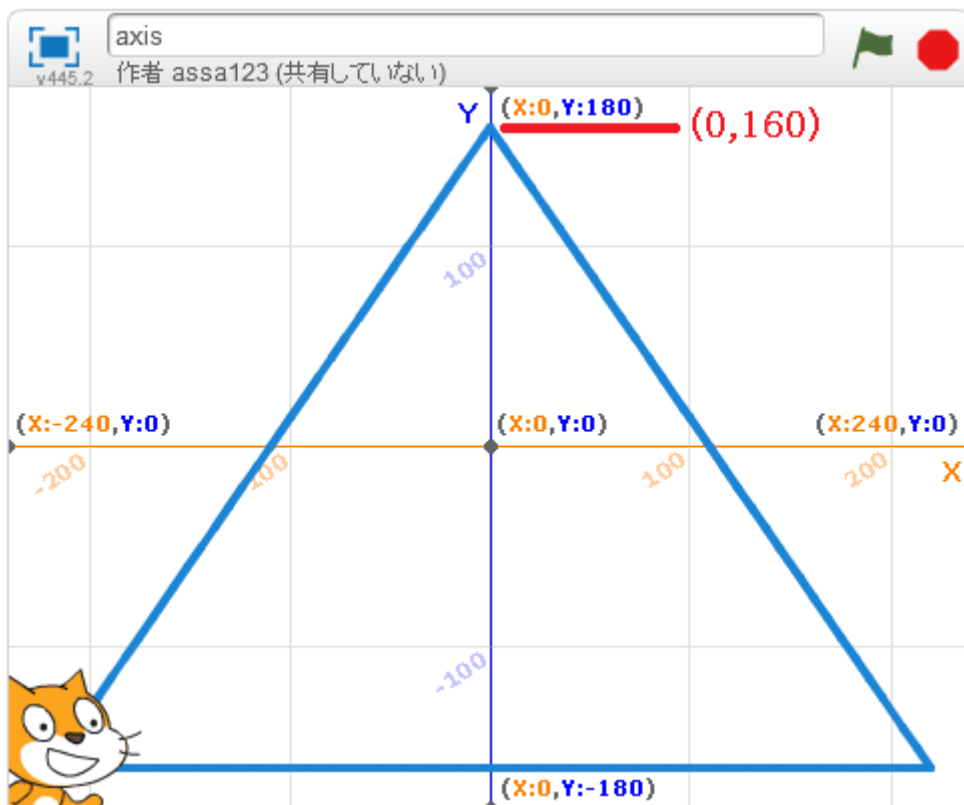


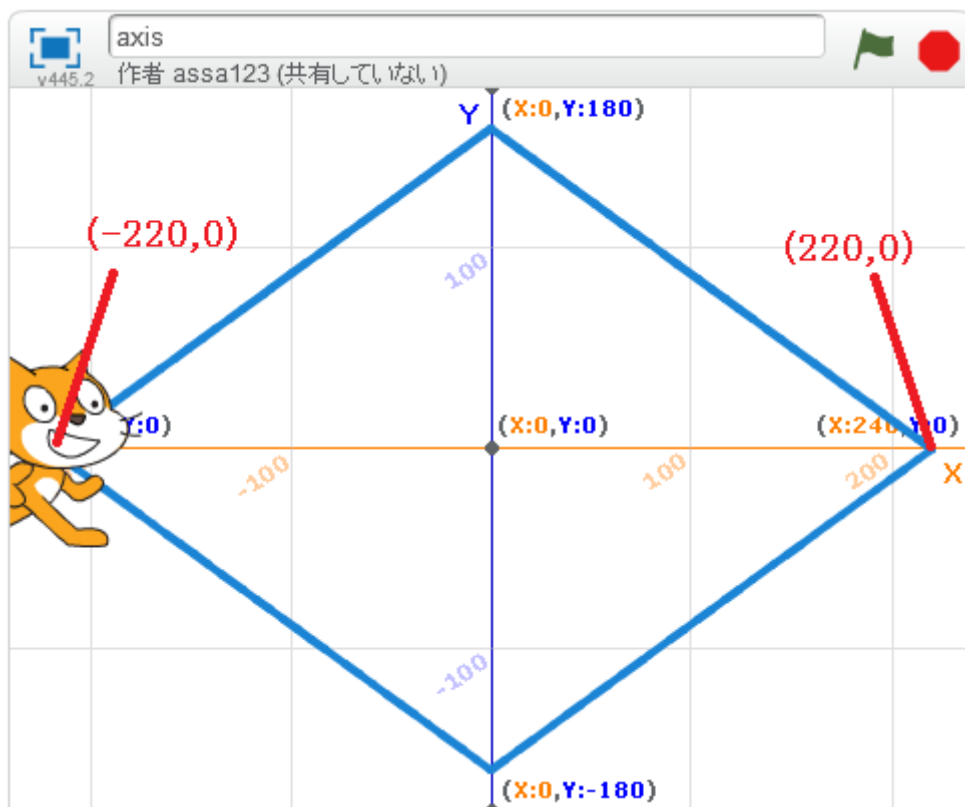
■ スクリプト



☆やってみよう

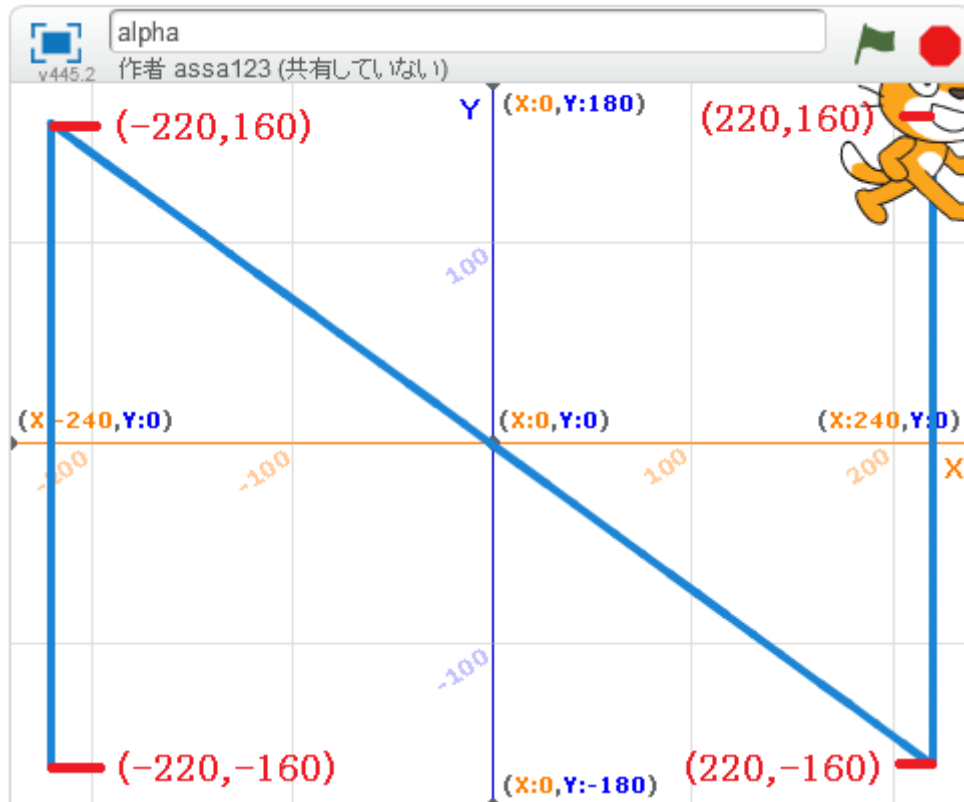
以下のように移動してみよう。



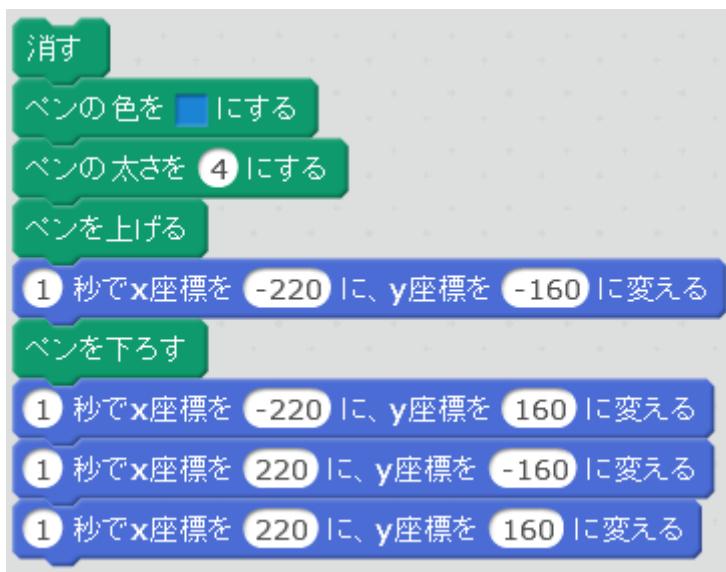


2. アルファベットを描く

■Nの文字を描く

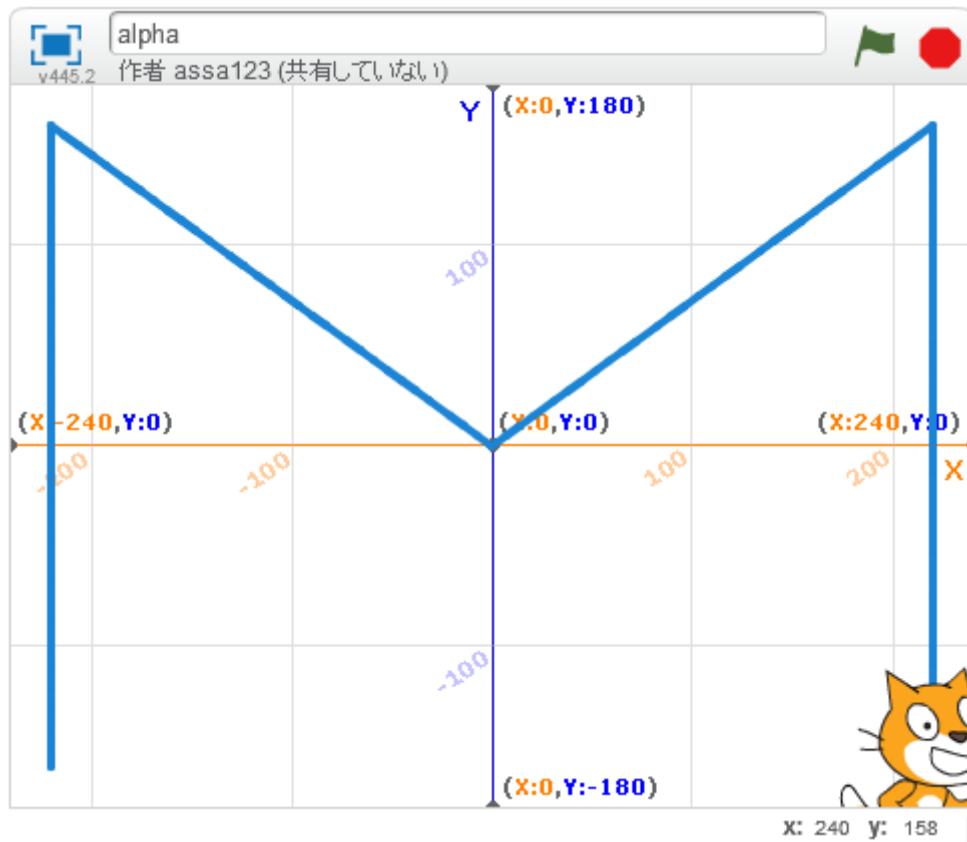


■スクリプト



☆やってみよう

- ・Mの文字を描いてみよう

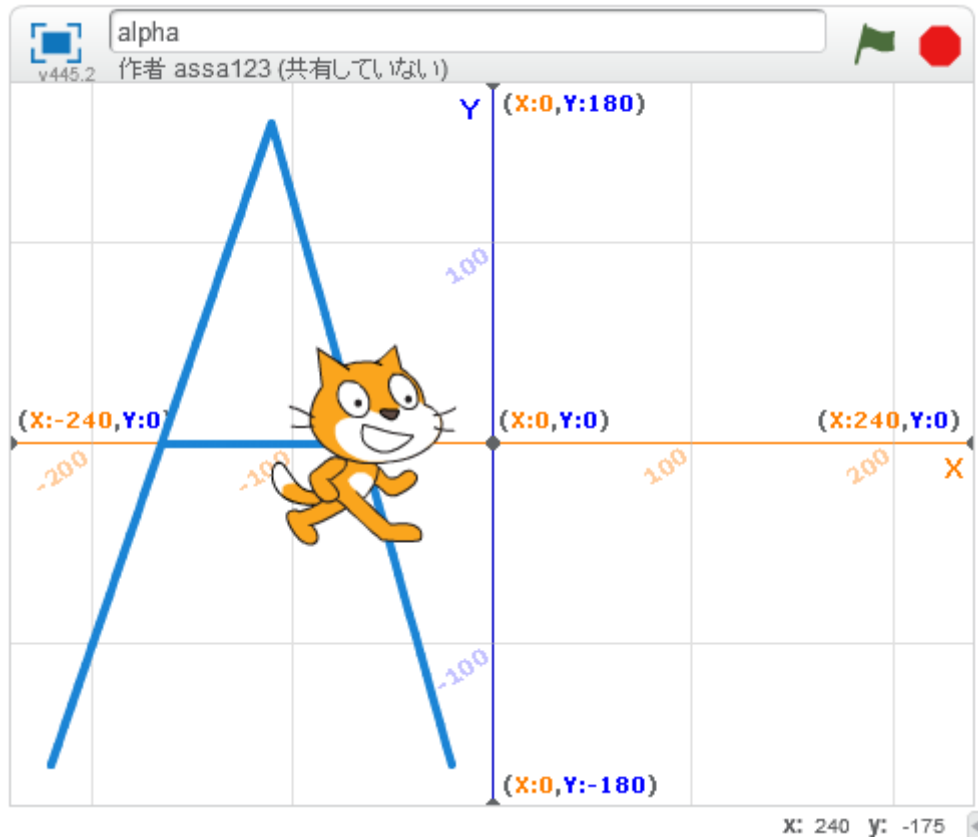


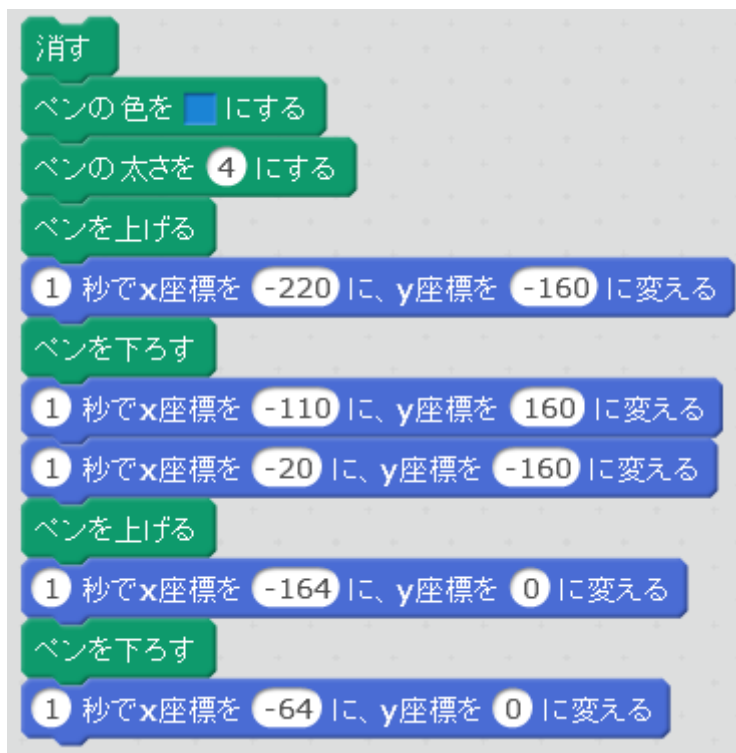
第5回 図形を描いてみようそのⅡ

1. 一筆書きできないアルファベットを描く

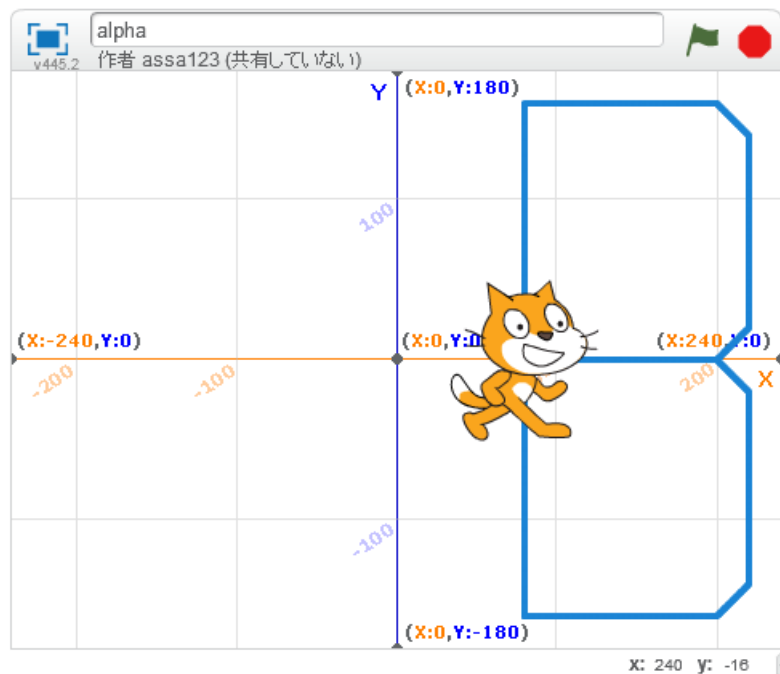
A や B は一筆書きできないのでいったんペンを上げて移動します。

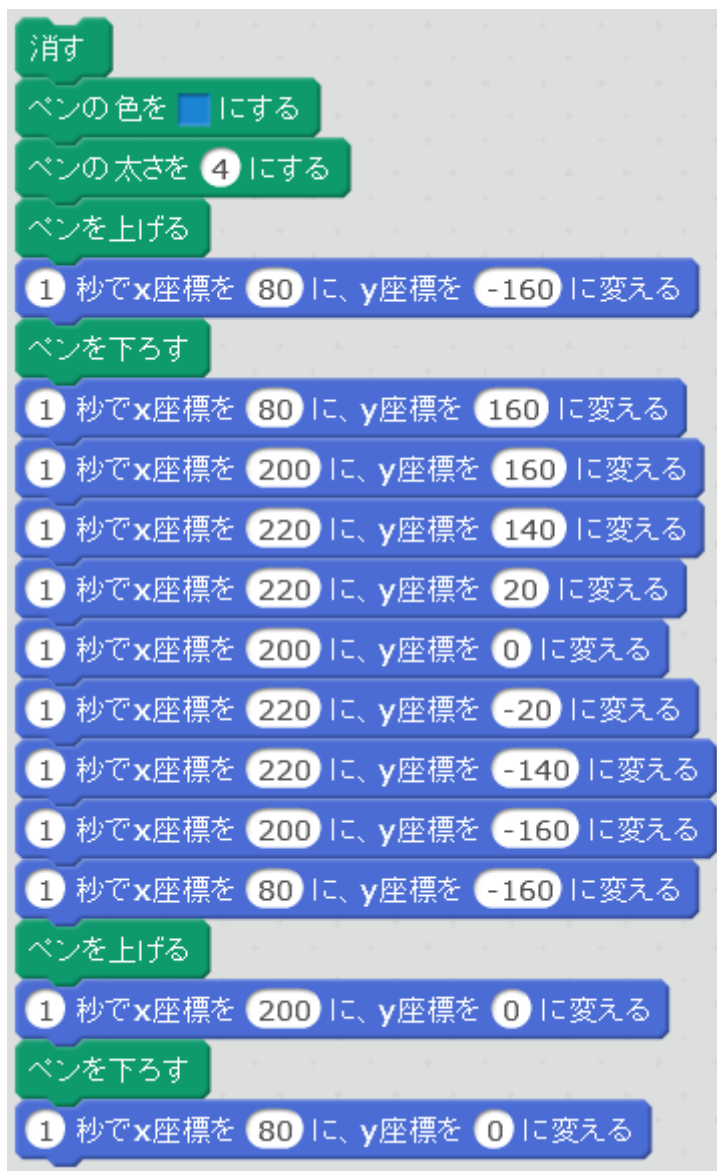
■ A の文字を描いてみよう





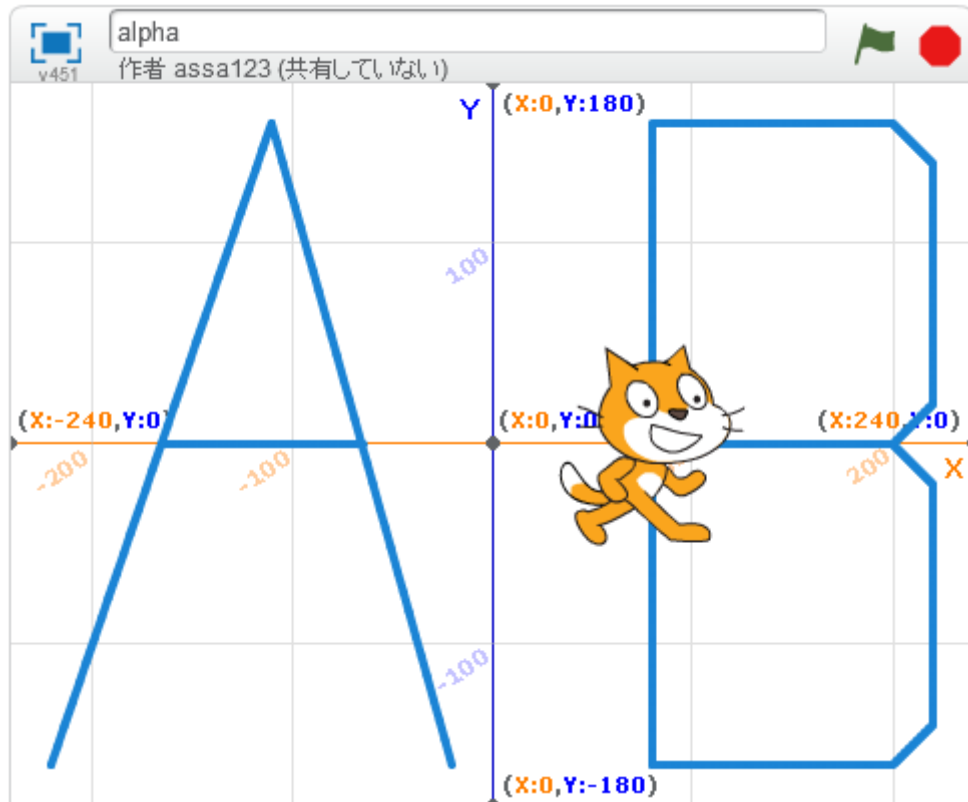
■ B の文字を描いてみよう

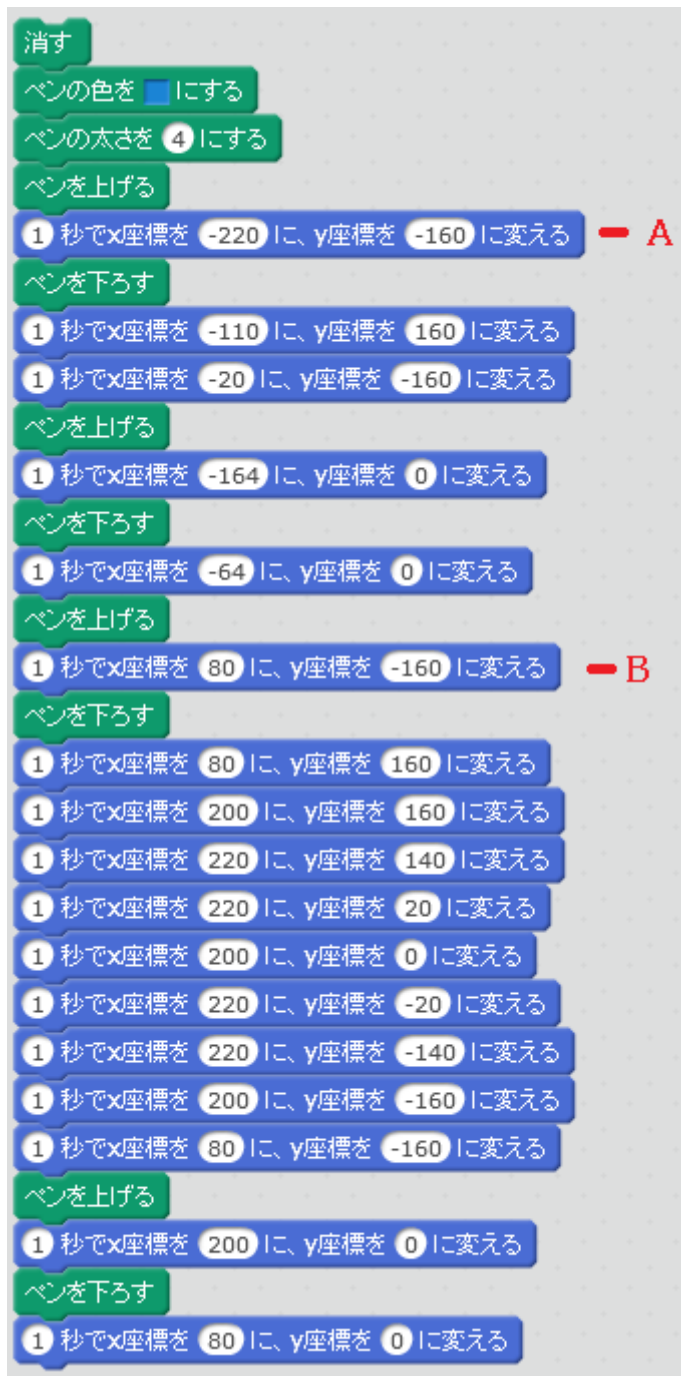




2. 複数の文字を書いてみよう

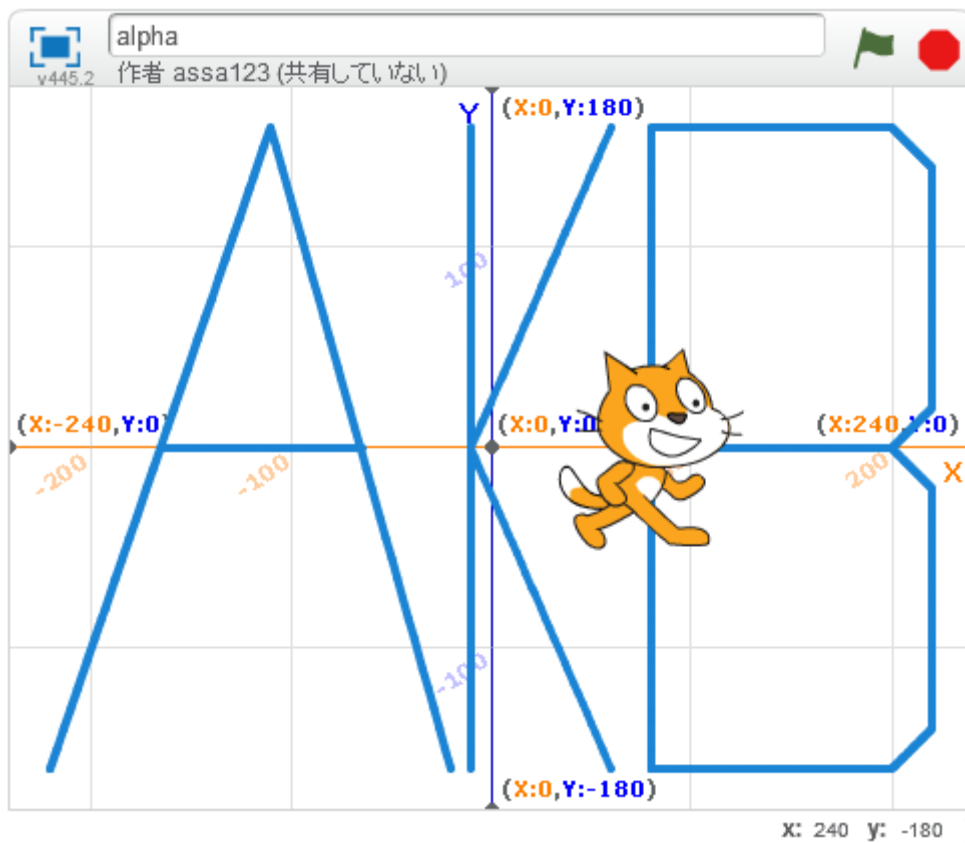
■A と B を2つ書いてみよう






☆やってみよう

- A、K、B の順に描きます。
- K のデータは自分で考えましょう。





☆やってみよう

A を書いたら  と 2 秒言う といってみよう。同じように K、B もいっていきよう。

第6回 プログラムのデータを管理する変数

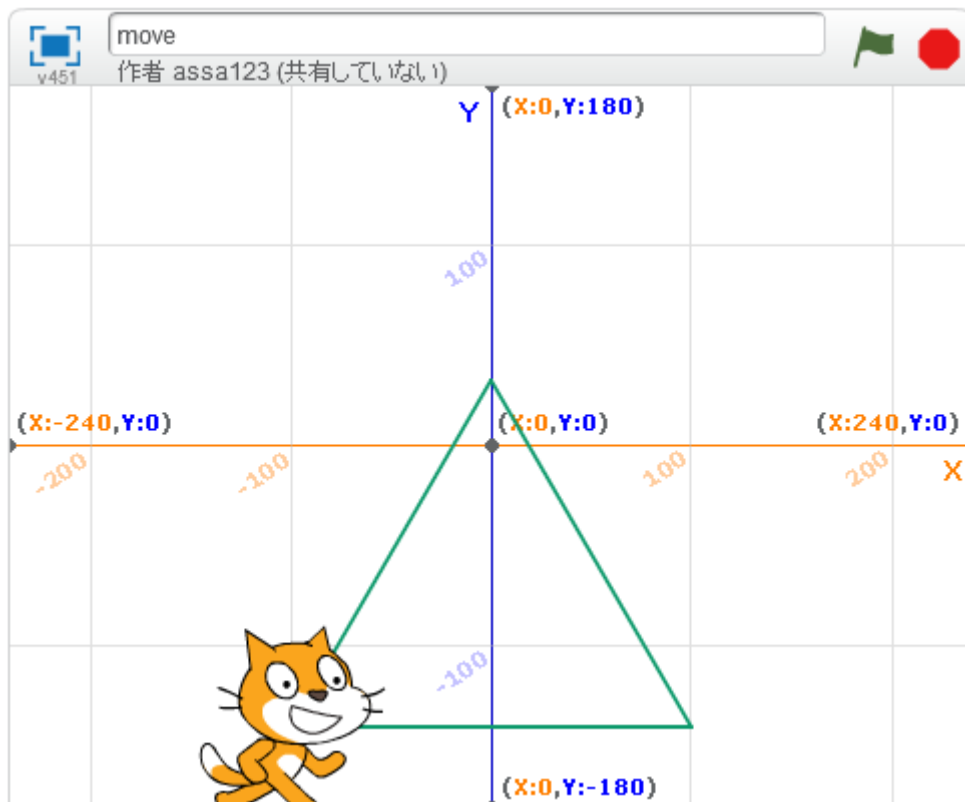
1. 3角形、4角形、5角形を描いてみよう

・3角形、4角形、5角形などを多角形と言います。

・  と  でスプライトを移動することで、様々な図形を簡単に描くことができます。

■3角形を描いてみよう

200歩移動、120度回転を3回繰り返せば三角形を描くことができます。スプライトの最初の位置は(-100,-140)の位置とします。



・スクリプト



☆やってみよう

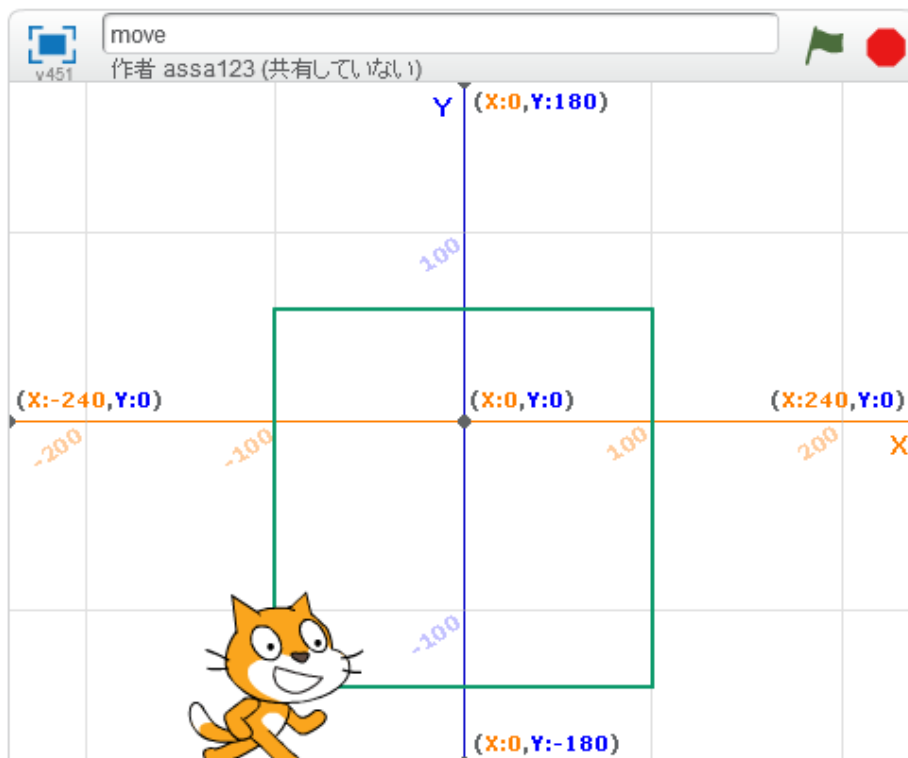
三角形を描いた時のプログラムの繰り返す回数と回転する角度を変えることにより四角形や五角形を描くことができます。



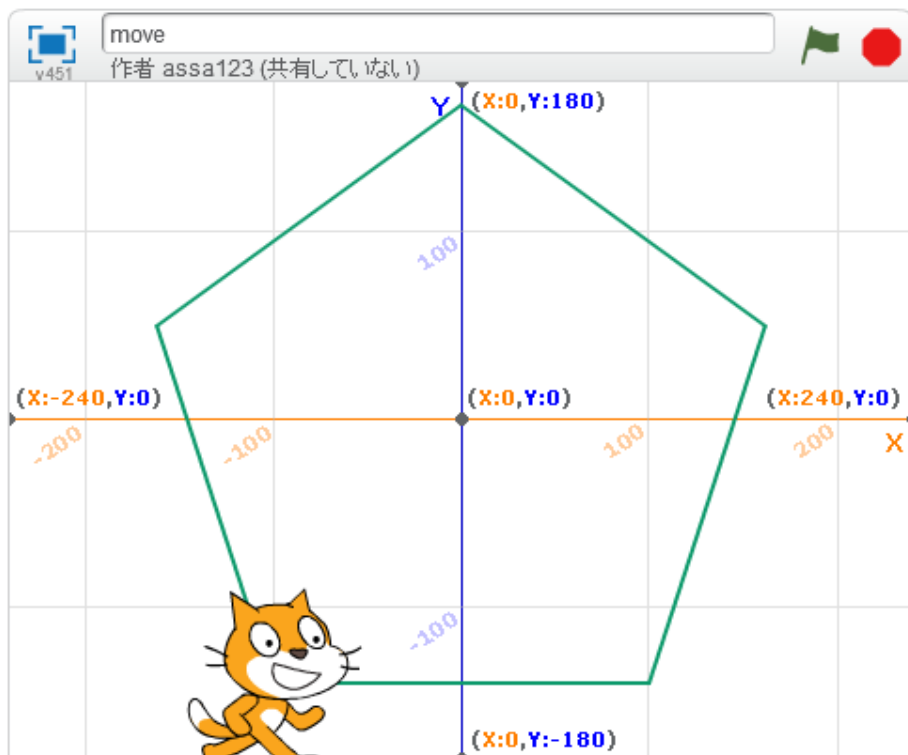
	繰り返す回数	回転する角度	
3 角形	3	120	$3 \times 120 = 360$
4 角形	4	90	$4 \times 90 = 360$
5 角形	5	?	$5 \times ? = 360$

■ 4 角形を描いてみよう

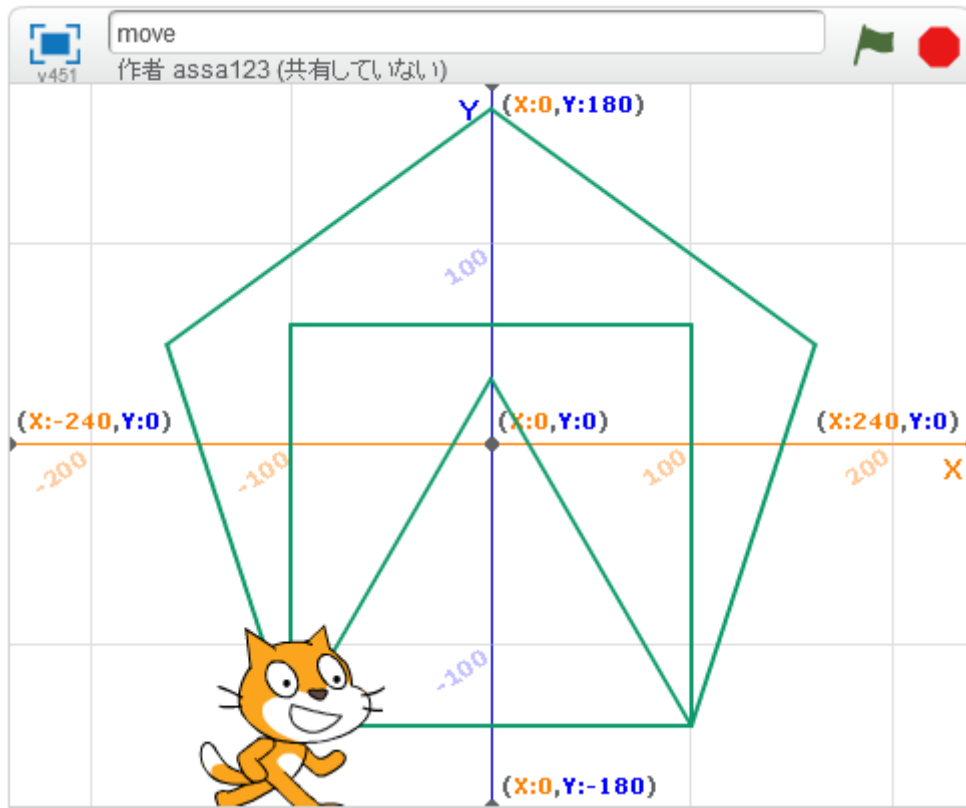
90 度回転しながら 4 回繰り返せば四角形が描けます。



■ 5 角形を描いてみよう



■ 3 角形、4 角形、5 角形を連続して描いてみよう



■ 複製（コピー）を作る

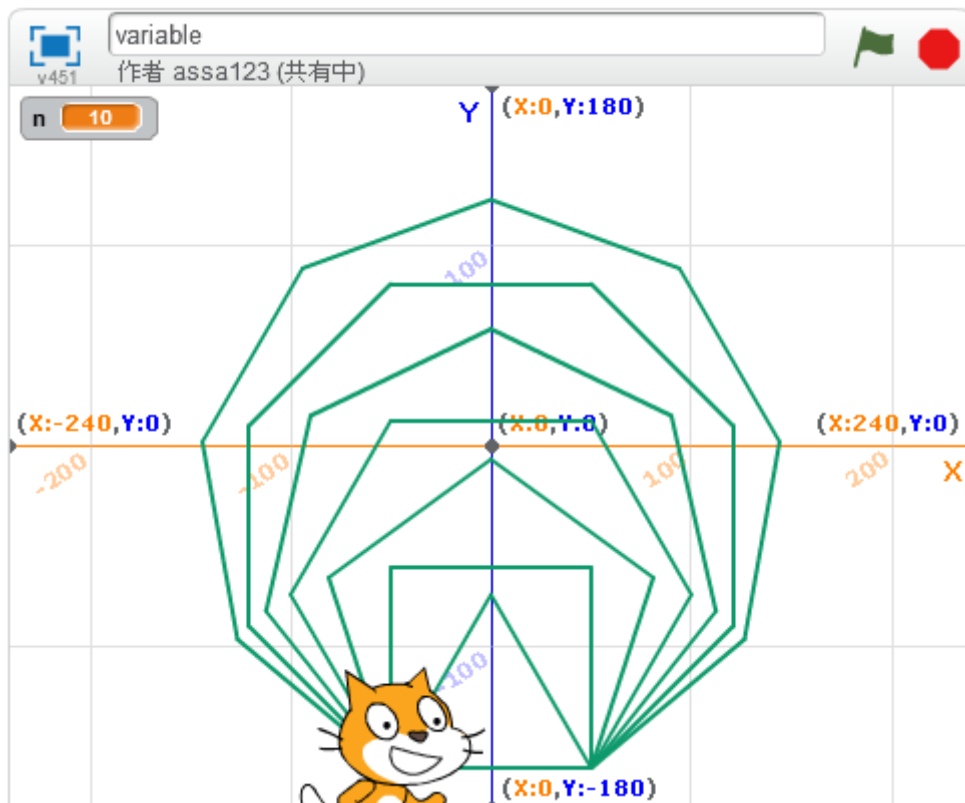
複製を作りたいブロックで右クリックし「複製」を選択します。



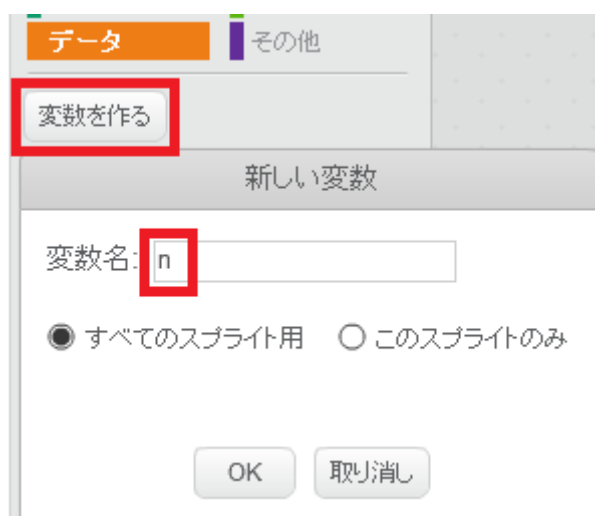


2. n 角形を描く

3 角形、4 角形、5 角形など個々の多角形を前に描きましたが、変数 n を使って n 角形を描いてみましょう。



■ 変数として n を作ります



- これで、以下のように変数と、変数を操作するスクリプトブロックが作成されます。



■変数の内容を変える

- 変数の内容を指定した内容にします。



- 変数の内容に指定した内容を加えた（減じた）値を新しい内容とします。





■計算式を作る

- 「スクリプト」の「演算」を選択します。黄緑色のスクリプトです。
- 足し算、引き算、掛け算、割り算ができます。



・ n 角形の場合の回転する角度は $360 / n$ で表せます。

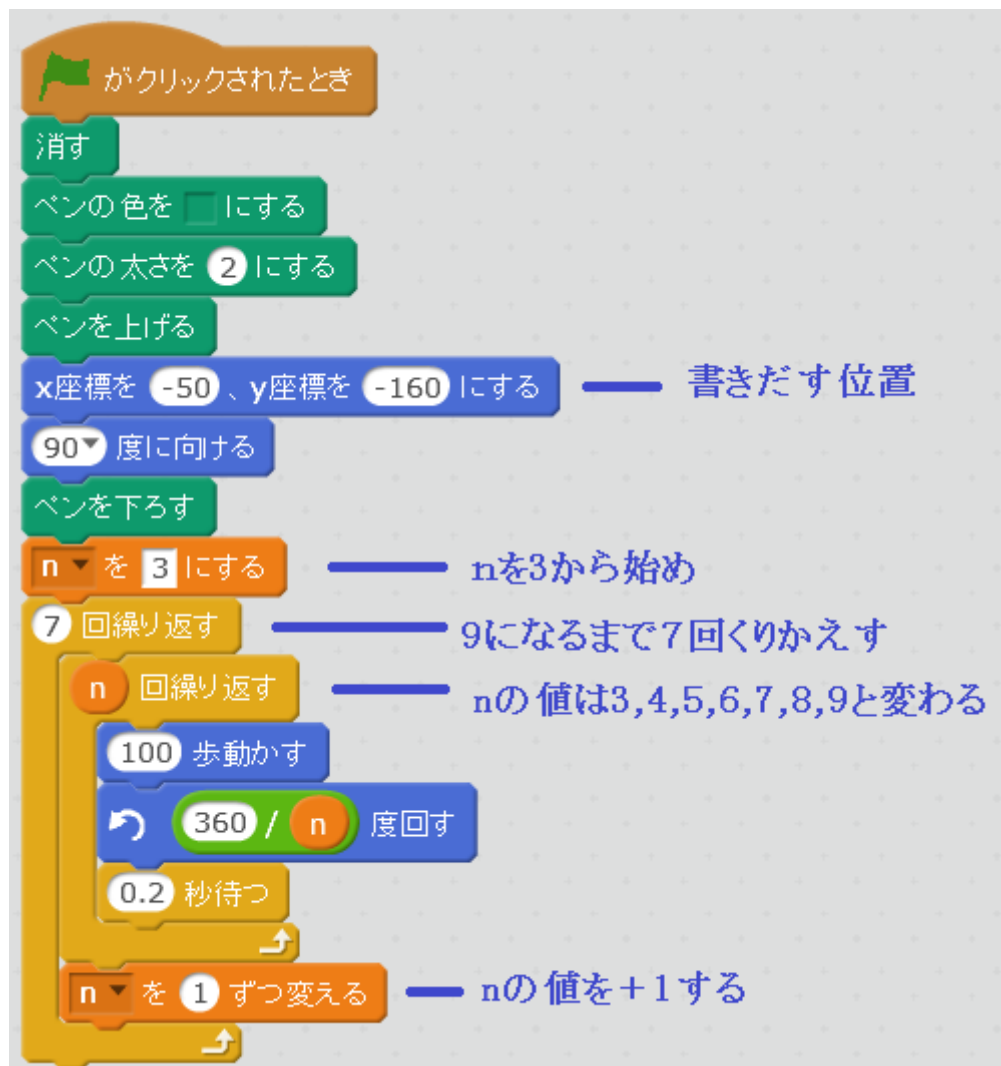
・  の中に数値や文字を入れる場合は直接入力します。変数を入れる場合は  をドラッグドロップします。



■ スクリプト

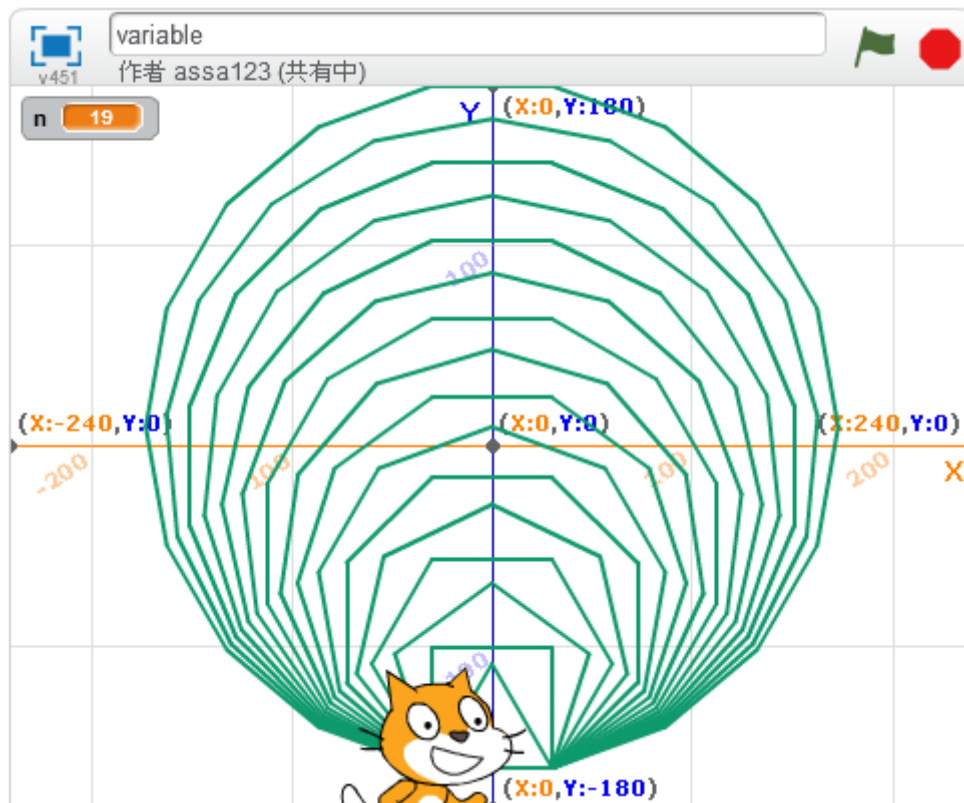
・ 書きだす位置を $(-50, -160)$ とします。

・ n を 3 から始めて 9 まで 7 回繰り返すことで 3 角形から 9 角形を描きます。



☆やってみよう

- 3 角形から 18 角形を描いてみよう
- 移動の長さは 60 とします。
- 図が画面の中央になるように開始の x 座標を考えましょう。
- 何回繰り返せばよいか考えましょう





「補足」変数の役割

- 変数とはプログラムで使用するデータを格納しておくためのものです。
- 変数には名前を付けます。これを変数名と呼びます。
- 一般のプログラミング言語では変数名には英字と数字だけを使い、「x」とか「x1」のような変数名を使います。
- Scratch にはこうした制限はありませんので、「合計」や「a&b」のような変数名も可能です。しかし本書では一般のプログラミング言語にならい、英数字の変数名を使用します。
- 英字の大小は区別されますので、「n」と「N」は別の変数になります。

第7回 データをまとめて管理するリスト

1. リストに音のデータを格納

■大きな栗の木の下での1小節の音符データをリストに格納

大きな栗の木の下で

イギリス民謡

♩=96

おお き な く り の き の し た で
あ な ー た と わ た し
な か よ く あ そ び ま しょう
おお き な く り の き の し た で


■音符の値

音符の値は以下のようにになっています。真ん中のドが60、レが62などです。



ド	レ	ミ	ファ	ソ	ラ	シ	ド
60	62	64	65	67	69	71	72

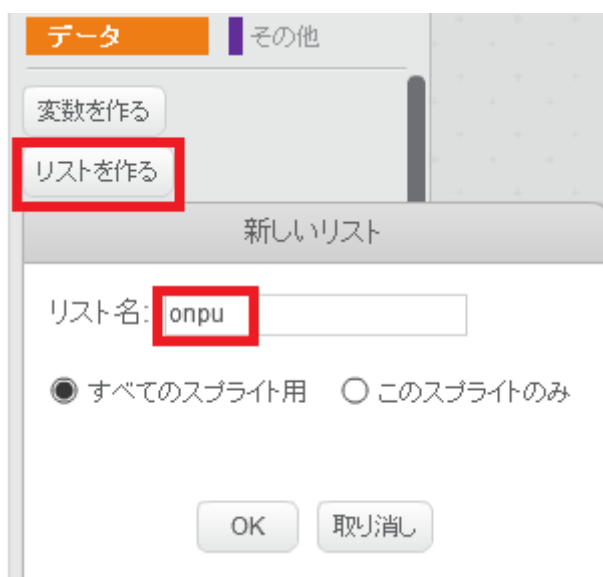
■拍数

四分音符  を1拍としたときの各音符の拍数は以下のようになります。

二分音符	2拍
八分音符	0.5拍
十六分音符	0.25拍

■リストの作り方

・「データ」ブロックの「リストを作る」で、オープンされたボックスにリスト名を入力します。



■ リストにデータを格納


リストの **onpu** と **haku** にそれぞれデータを格納します。

 をクリックします。空欄に値を入力し、「Enter」キーを押し、次のデータを入力します。

	ド	ド	レ	ミ	ミ	ソ	ミ	ミ	レ	レ	ド
onpu	60	60	62	64	64	67	64	64	62	62	60
haku	1	0.5	0.5	0.5	0.5	1	0.5	0.5	0.5	0.5	2


onpu

1

 長さ: 1


onpu

1 60
2 60
3 62
4 64
5 64
6 67
7 64
8 64
9 62
10 62
11 60

 長さ: 11


haku

1

 長さ: 1

haku

1 1
2 0.5
3 0.5
4 0.5
5 0.5
6 1
7 0.5
8 0.5
9 0.5
10 0.5
11 2

 長さ: 11

2. リストのデータを演奏

■楽器を音符で鳴らす

楽器を鳴らすスクリプトブロックとして以下があります。特別に音を登録しなくても音が鳴ります。

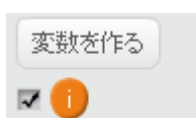
- ・鳴らす楽器を 1（ピアノ）～21（シンセバット）から選びます。



- ・指定した楽器で指定した音符の音を指定拍数で鳴らします。



■リストのデータを取り出すための変数として i を作ります



■リストのデータを取り出す

リスト **onpu** の **i** 番目のデータは以下のように取り出すことができます。



リスト **haku** の **i** 番目のデータは以下のように取り出すことができます。



■スクリプト

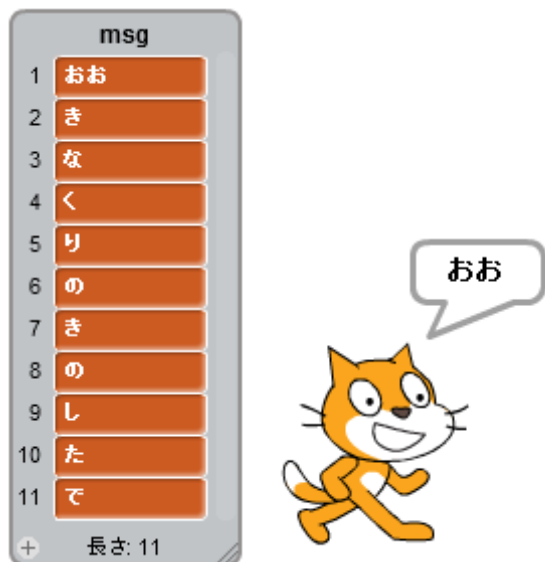


☆やってみよう

・音の演奏に合わせて「おお き な く り の き の し た で」と「ねこ」に話させましょう。

・データをリスト **msg** に格納します。

・リスト **msg** の **i** 番目のデータを取り出すにはどうしたらよいでしょうか。



「補足」リストの役割

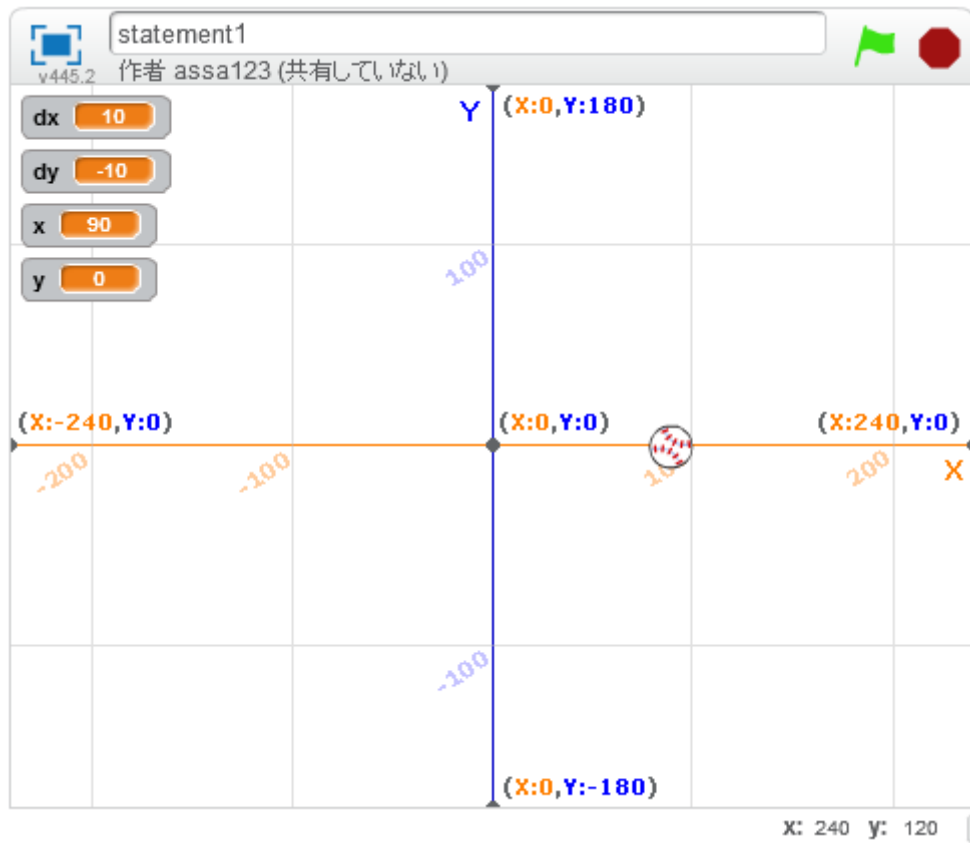
・変数には1つのデータしか入りませんが、リストには多くのデータを格納することができます。

・リストのデータを変数を使って取り出すことでプログラミングがし易くなります。

・一般のプログラミング言語ではリストのことを配列と呼んでいます。

第8回 プログラムを制御するその I


1. ボールを変数を使って跳ね返す

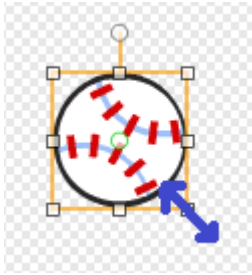



■ スプライトを Baseball にします



■ ボールの縮小とセンタリング

- ・ 隅の  にマウスを合わせてドラッグすることでコスチュームを縮小します。ボールのサイズは半径10程度にしてください。



- ・  を選択し、ボールの中央をクリックします。

これでボールの中心がセンタリングされます。



■変数として x,y,dx,dy を作ります



■ボールを横に進めるには

- ・ ボールの位置を変数 x で管理し、移動量を変数 dx で管理します。 dx の値が 10 なら右に進み、-10 なら左に進みます。

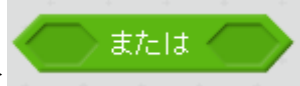


■ 条件式

ボールが左の壁に当たったと判定する式は



ボールが右の壁に当たったと判定する式は



この2つに式を でつなげます。

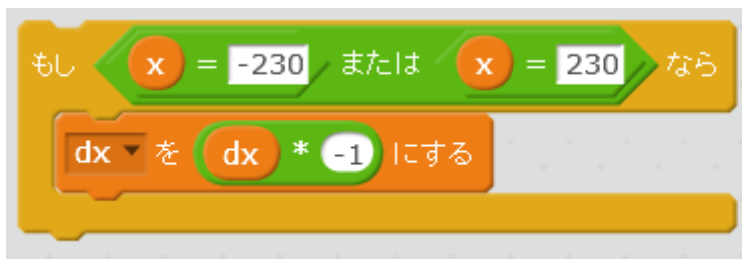
壁の x 座標の判定が「240」でなく「230」となっているのはボールの半径が 10 なら、ボールの中心が 230 に来た時が壁に当たったと判定できるからです。

■ 条件判定

条件判定には以下のスクリプトを使います。



・左右の壁に当たるたびに dx の符号を反転することでボールを跳ね返すには以下のようにします。



■ スクリプト

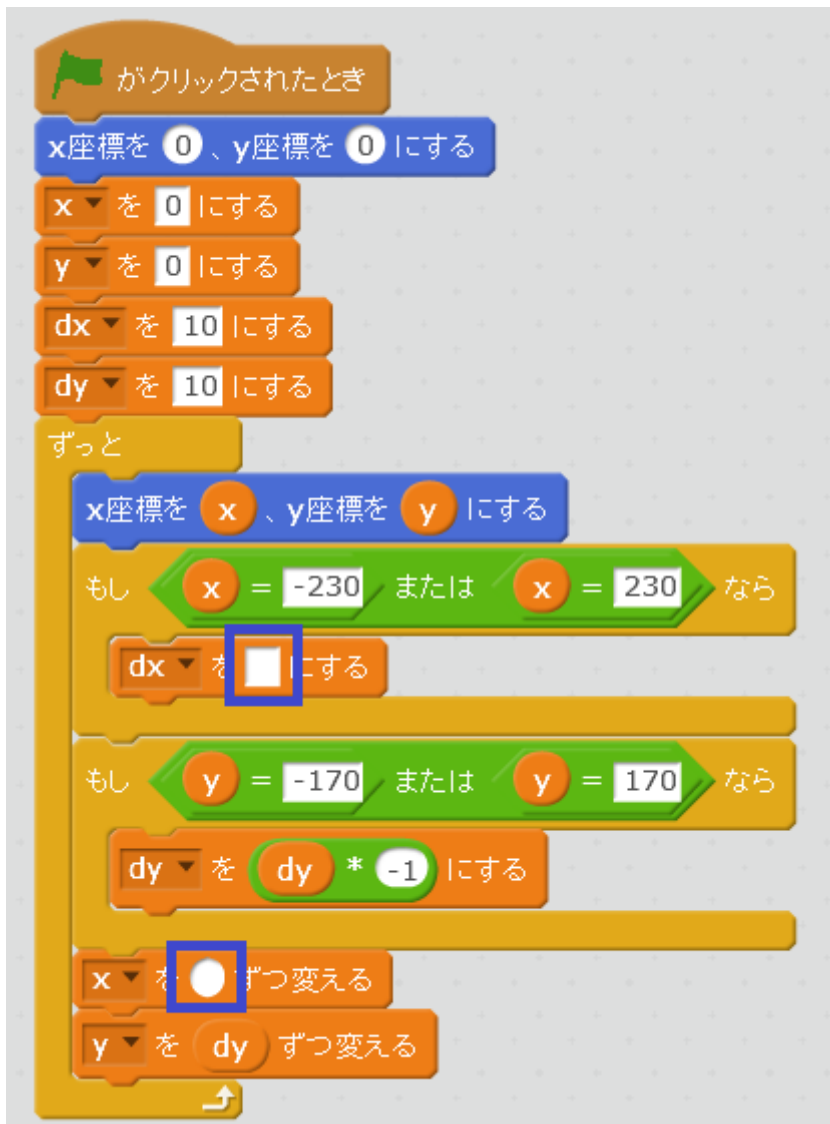


☆やってみよう

- ・ボールを縦に移動して壁に当たったら跳ね返すようにしましょう



- ・ ボールを斜めに移動して壁に当たったら跳ね返すようにしましょう



「補足」制御スクリプト

プログラムは「繰り返し」、「条件判定」などの制御スクリプトを使ってプログラムの流れを制御します。

■ 繰り返し

・ 繰り返しを行うスクリプトブロックとして以下があります。「ずっと繰り返す」と「指定回数繰り返す」はすでに説明しました。

・ 「条件になるまで繰り返す」は「7. 同じ処理をひとつにまとめるブロック」で説明します。

・ ずっと繰り返します



・ 指定した回数だけ繰り返します



・ 指定した条件になるまで繰り返します



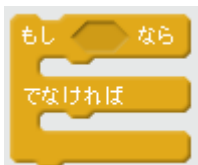
■ 条件判定

条件判定を行うスクリプトブロックとして以下があります。


・ もし条件を満たしたらブロック内を実行します



・ もし条件を満たしたら上のブロックを、そうでなければ下のブロックを実行します



「補足」条件式

 に入れる条件式として以下があります。

- 左辺は右辺より小さい



- 左辺は右辺より大きい



- 左辺と右辺は等しい



条件が2つ以上ある場合は以下を使います

- 左と右の条件がともに満たすとき



- 左または右の条件式のどちらかを満たすとき



第9回 プログラムを制御するそのⅡ

1. ボールをラケットで跳ね返す



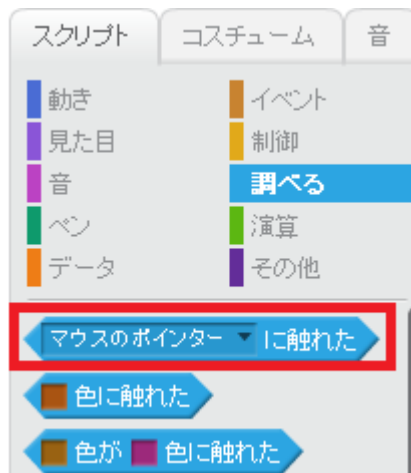
■ スプライトに Paddle を追加します



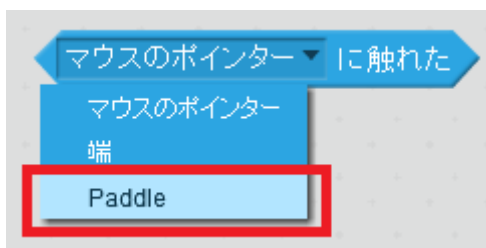
■ 変数として x,y,dx,dy を作ります



■パドルに当たったかどうか調べるには
「調べる」の「～に触れた」を使います。



スクリプトをスクリプトエリアに移してから「Paddle」を選択します。



■パドルをマウスの移動に合わせて移動するには
「調べる」の「マウスの x 座標」を使い、Paddle の位置を移動します。Paddle の y 座標は
「-140」固定です。これでマウスの移動に合わせてパドルが水平に移動します。

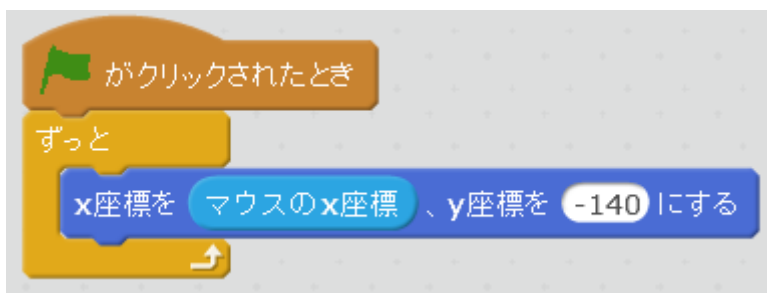


■ スクリプト

・ Baseball

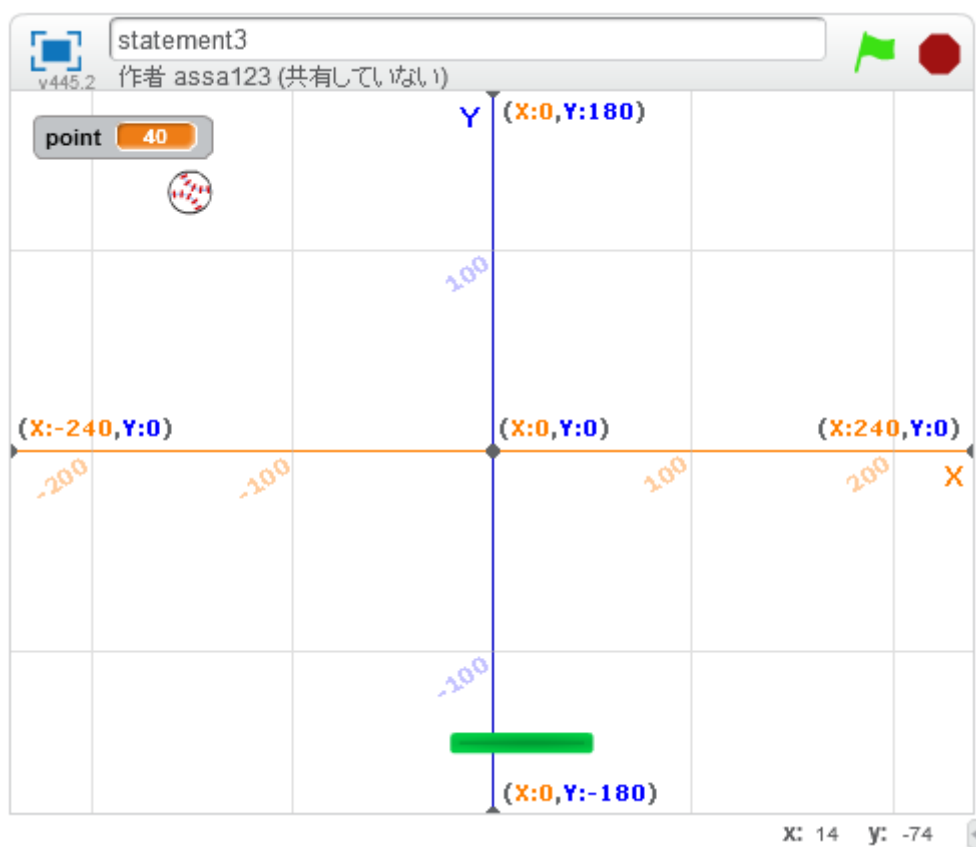


• Paddle



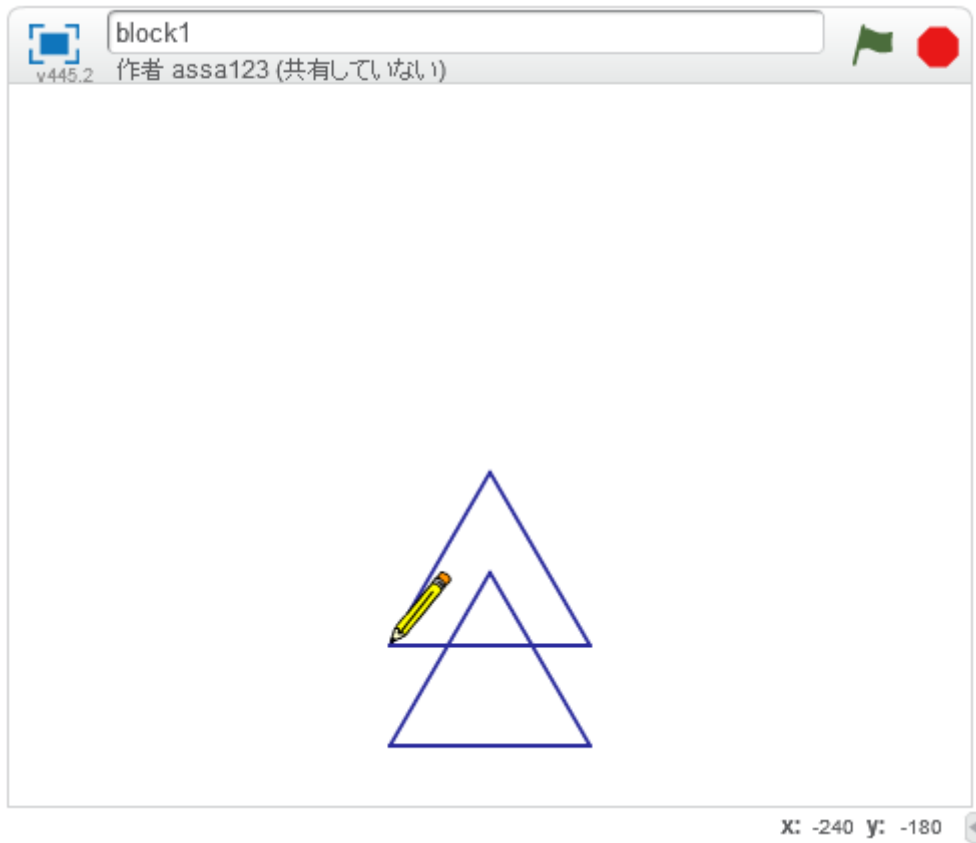
☆やってみよう

- ボールがラケットに当たるたびにポイントを+10しながら、ポイントを表示してみよう。変数として **point** を作り、そこにポイントを加算して行きます。



第10回 同じ処理をひとつにまとめるブロック

1. 3 角形を描くブロック triangle を作ってみよう



■ブロックの作り方

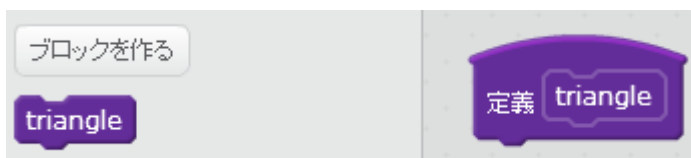
- ・「その他」ブロックの「ブロックを作る」をクリックします。



- ブロックの名前を入力します。ここでは「triangle」としました。



- 以下のように **triangle** という名前の新しいブロックが作られます。

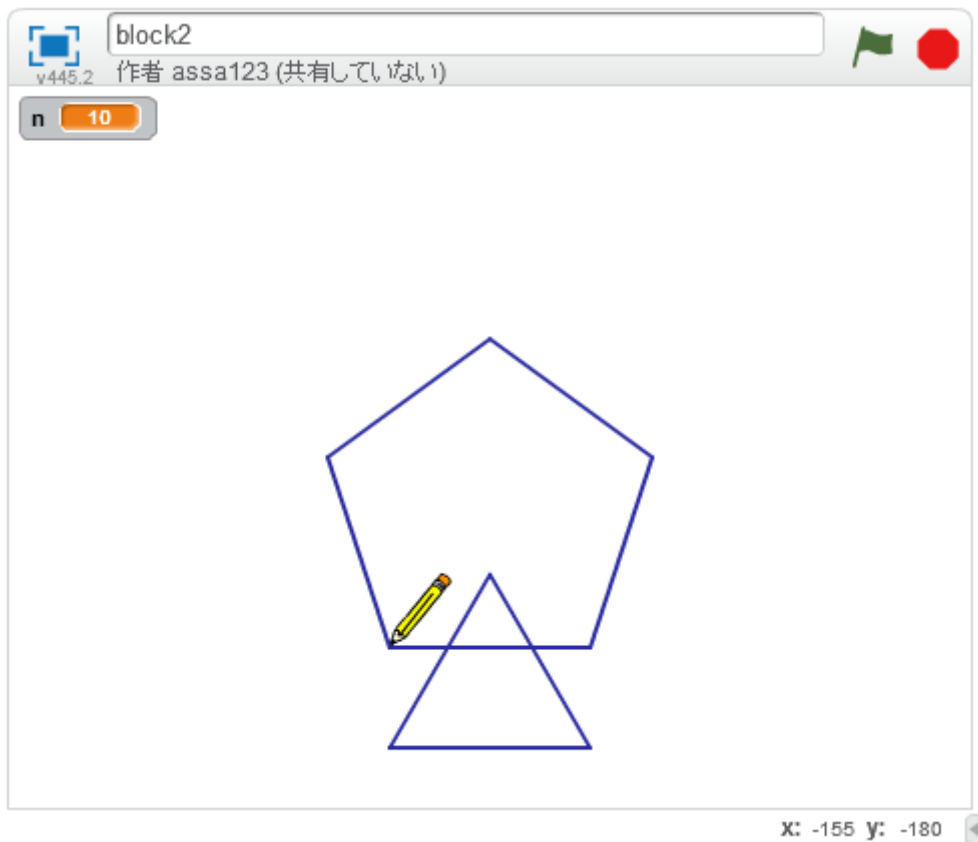


■ スクリプト





2. n角形を描くブロック polygon を作ってみよう



■ブロックに引数を指定

- ・「オプション」を選択し、「数値の引数を追加」を選択します。
- ・引数として「n」を入力します。

新しいブロック

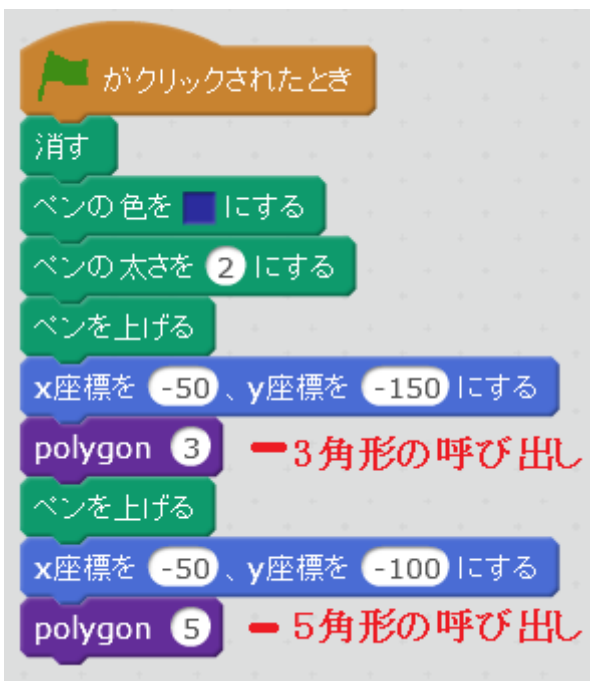


- 以下のように **polygon** という名前の新しいブロックが作られます。

n が引数で、呼び出し元からデータを受け取ります。



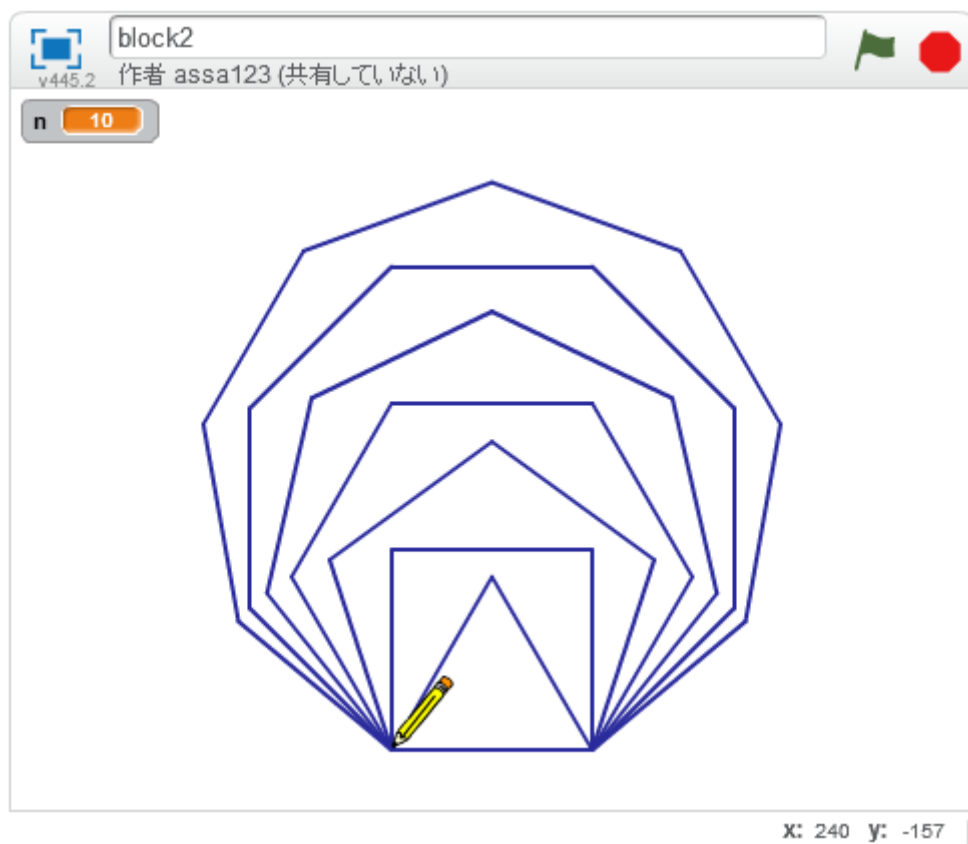
■ スクリプト





☆やってみよう

- polygon を使って 3 角形から 9 角形を描いてみよう
- 変数として n を作ります。





「補足」ブロックの役割

- ・同じ処理を何度もする場合に、この処理をブロックとして定義し、必要に応じて呼び出すようにするとプログラムを効率的に書くことができます。
- ・自分で定義したスクリプトブロックは紫色です。
- ・新しく定義したブロックには呼び出す方からデータを与えることができます。これを引数と言います。
- ・新しく定義したブロックは一般のプログラミング言語では関数とかプロシージャとかメソッドなどと呼ばれているものです。

「補足」引数

- ・新しく定義したスクリプトブロックを呼び出す際に、データを渡すことができます。
- ・これを引数と言います。
- ・引数は変数と似た動作を行いますが、定義されたブロックの中だけに固有のものです。
- ・引数はブロックの作成時の「オプション」を選ぶことで作成できます。引数として渡すデータが数値、文字列、真偽値で異なる引数ブロックを選びます。引数に名前を付けます。

3. リカーシブ・グラフィックス

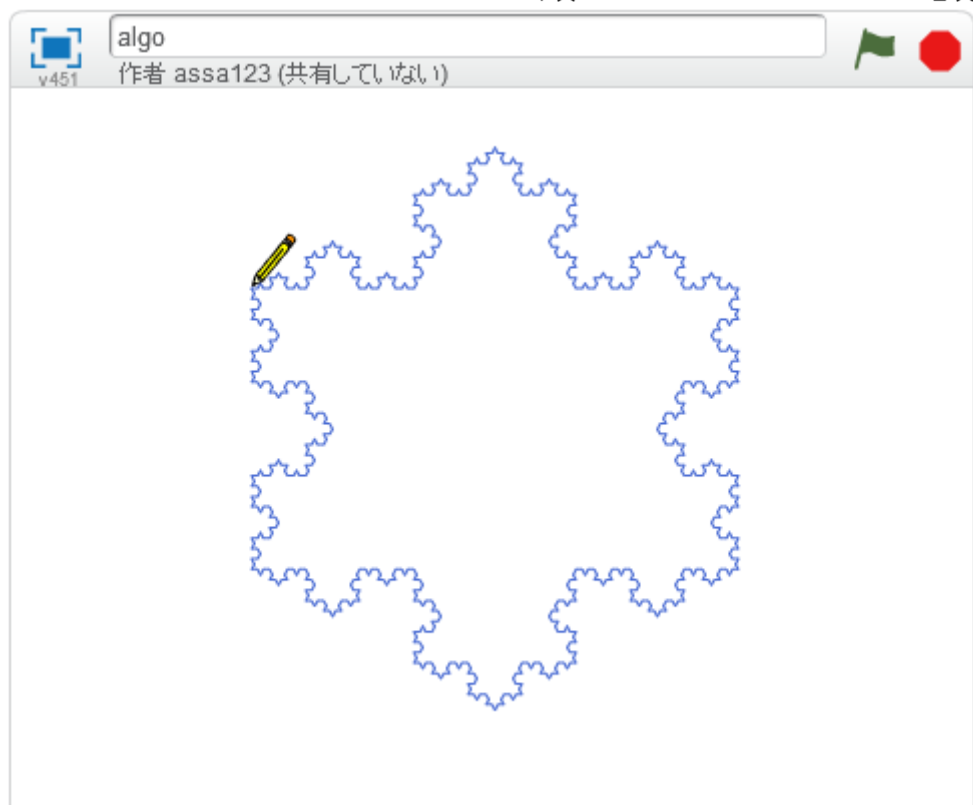
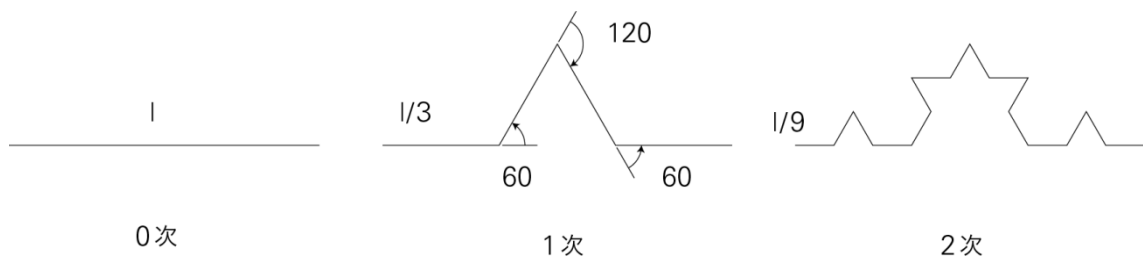
リカーシブ・グラフィックスはグラフィックスの世界を解析的に表現せずに、再帰的に表現しようとするものです。再帰を使うと自然に近い図形（入り組んだ海岸線や樹木）がいつも簡単に表現できます。

■ コッホ曲線

コッホ曲線は、数学者のコッホにより発見されたものです。コッホ曲線は以下のように定義されています。

0 次のコッホ曲線は長さ 1 の直線である。1 次のコッホ曲線は、1 辺の長さが $1/3$ の大きさの正三角形形状のどっぴりを出す。2 次のコッホ曲線は、1 次のコッホ曲線の各辺（4 つ）に対し、1 辺の長さが $1/9$ の大きさの正三角形形状のどっぴりを出す。

コッホ曲線



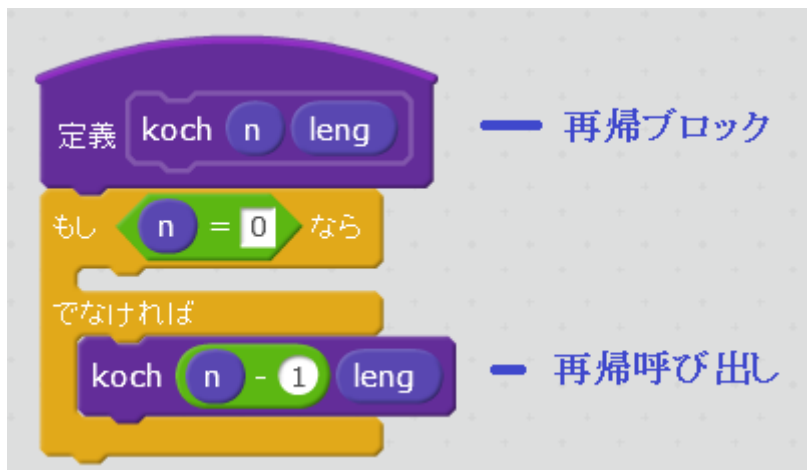
■koch ブロックの作成

引数を n と $leng$ とします。



■再帰ブロックとは

ブロックの中から再び同じブロックを呼び出すようなものを再帰ブロックといいます。

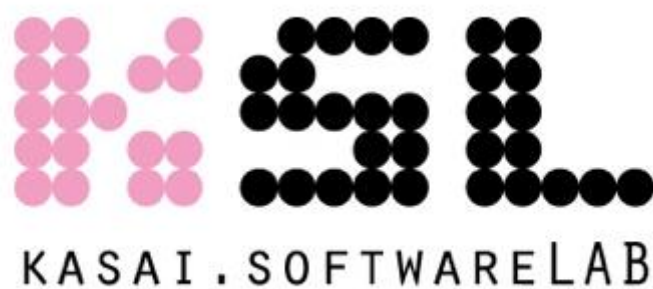


■ スクリプト

n 次のコッホ曲線を描く手順は以下です。

- ① $n-1$ 次のコッホ曲線を 1 つ描く。
- ② 向きを 60° 変えて $n-1$ 次のコッホ曲線を 1 つ描く。
- ③ 向きを -120° 変えて $n-1$ 次のコッホ曲線を 1 つ描く。
- ④ 向きを 60° 変えて $n-1$ 次のコッホ曲線を 1 つ描く。





Scratch でプログラミングをたのしもう！

2016 年 4 月 25 日 初版 第 1 刷

©2016 河西 朝雄

本書は学校等の授業で使用する場合を除き、本書の一部または全部を著作権法の定める範囲を超え、無断で複製、複製、転載、あるいはファイルに落とすことを禁じます。

本書に記載された内容は、情報の提供のみを目的としています。したがって、本書を用いた運用は、必ずお客様自身の責任と判断によって行ってください。これらの情報の運用の結果について、発行者および著者はいかなる責任も負いません。