



プログラム入門

目次

1. はじめに	2
1.1. プログラムとは	2
1.2. プログラミングとは	2
1.3. SCRATCH とは	2
1.4. 各研修とのつながり	2
2. Scratch の基本操作	3
2.1. まずは動かしてみよう	3
2.2. 画面構成	6
2.3. 操作方法	7
3. 「動き」に関するブロック	12
4. 「見た目」に関するブロック	14
5. 「変数」に関するブロック	16
6. 「制御」に関するブロック	21
7. 「演算」に関するブロック	25
8. 「イベント」に関するブロック	27

1. はじめに

1.1. プログラムとは

プログラムとは物事を行う手順や予定といった意味の英単語になり、ITの世界ではコンピュータが行うための処理を記述したものをプログラムといいます。

日常生活では様々なところでプログラムが組まれたものが動いています。例として通販サイトなどのWebサイトやスマートフォンのアプリなどが該当します。

1.2. プログラミングとは

プログラミングとはプログラムを構築する作業のことをいいます。

プログラムを構築する際にはプログラミング言語というものを使用します。プログラミング言語には種類があり、本資料ではScratch（スクラッチ）と呼ばれる言語を使用していきます。

1.3. Scratch とは

Scratch は、積み木のようなブロックを組み合わせて「プログラム」を作成できるプログラミング言語です。一般的なプログラミング言語のような難しい部分を排除し、分かりやすく、楽しくプログラミングを勉強できます。また、Scratch を使ってアニメーションやゲームも制作できます。制作した作品はインターネット上で世界中の人と共有でき、他の人が作った作品に触れることも可能です。

1.4. 各研修とのつながり

1.4.1 Java 研修

Java 研修では、Java と呼ばれるプログラミング言語を使用してプログラミングを学んでいきます。言語は違いますがプログラミングをするという点では同じように考えることができます。

1.4.2 組込み研修

組込み研修では、C と呼ばれるプログラミング言語を使用してプログラミングを学んでいきます。言語は違いますがプログラミングをするという点では同じように考えることができます。

1.4.3 インフラ研修

インフラ研修では、シェルスクリプトと呼ばれる言語を使用して OS 操作やファイル操作を行うことがあります。その時にプログラムを組む考え方などが必要になってきます。

2. Scratch の基本操作

2.1. まずは動かしてみよう

まずは Scratch を動かしてみましょう。Scratch を使用するためには、以下の条件を満たした PC が必要です。

- インターネットに接続されている
- Web ブラウザがインストールされている

※Scratch 公式サイトでは、オンライン版の Scratch の他に、PC にインターネット接続なしで使用できる「Scratch オフラインエディタ」もあります。当研修では、公式サイトで公開されているサンプル作品を利用するため、オンライン版を使用します。

※ブラウザは一部非対応のものが 있습니다。当資料では Microsoft Edge、Google Chrome を想定しています。

手順 1：Web ブラウザを開き、Scratch の公式サイトにアクセスします。

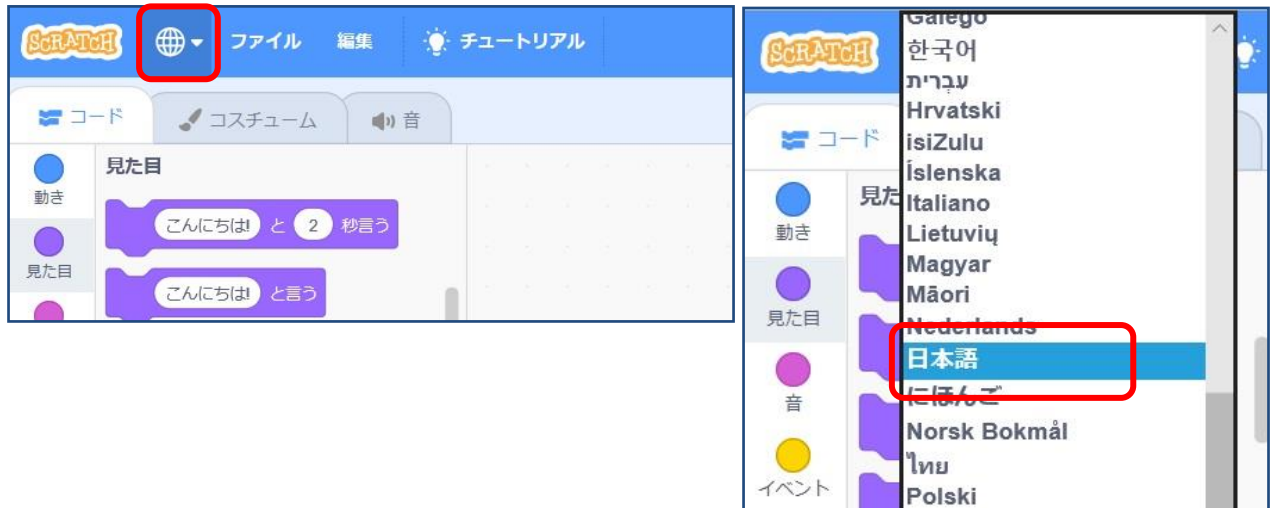
URL： <https://scratch.mit.edu/>

手順 2：画面左上のリンク「作る」をクリックします。



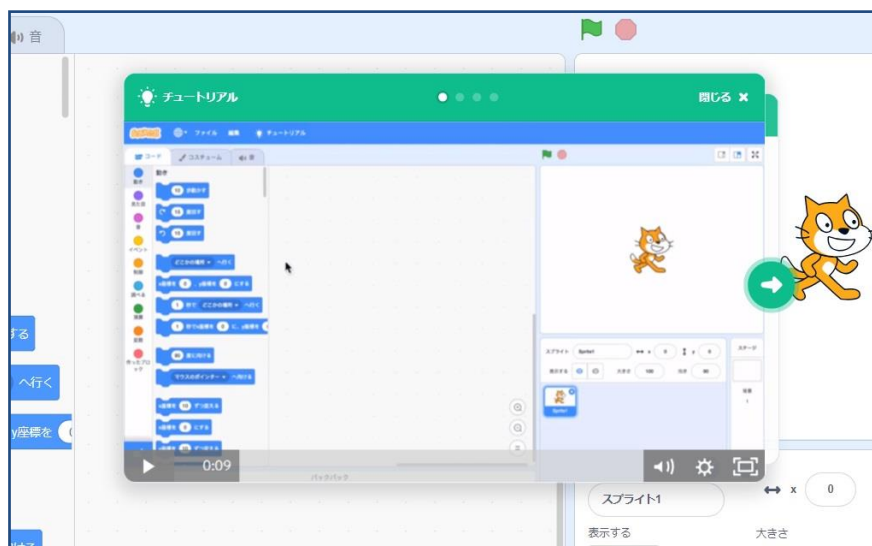
※日本語表示ではない場合のみ、以下の手順を実行します

下記のマークをクリックし、下までスクロールして「日本語」を選択します。



ここまで終わったら、Scratch でプログラムをする準備は完了です。次は、試しにプログラムを動かしてみしましょう。

※最新版 Scratch ではチュートリアル動画を閲覧できるようになりました。とてもわかりやすいので、時間があれば、ご覧になってください。



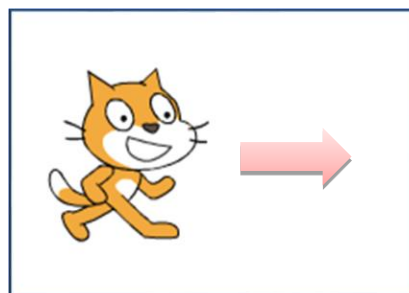
手順3：「ブロックパレット」から「スクリプトエリア」へ、
ドロップして配置します。



ブロックをドラッグ・



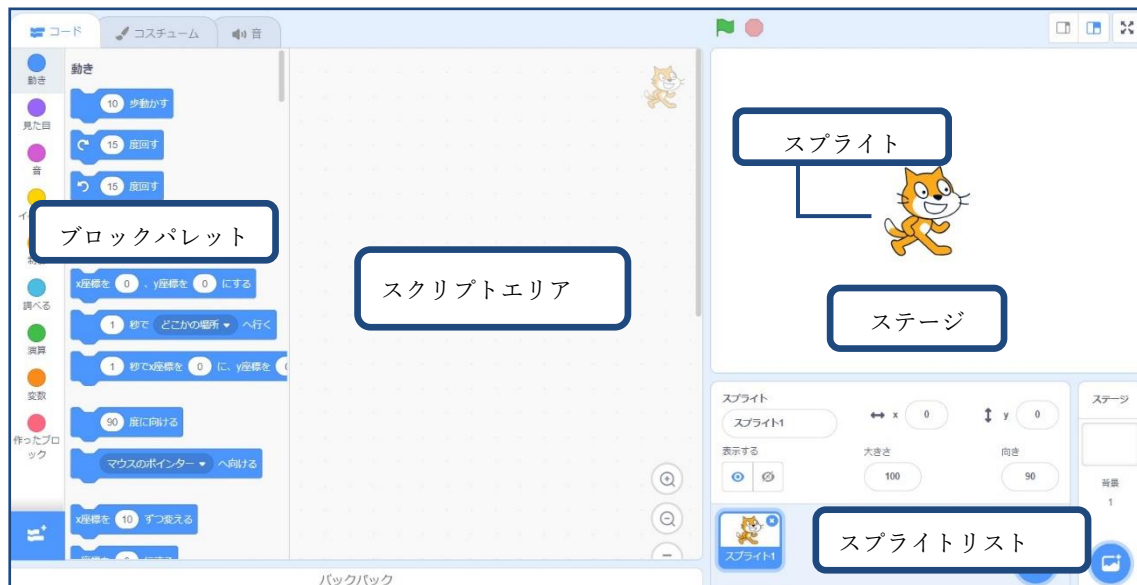
「スクリプトエリア」に配置されたブロックをクリックすると、猫のキャラクターが右に移動します。



このように、Scratch のプログラムは簡単に動かすことが可能です。次からは、Scratch の画面構成や基本的な機能を紹介します。

2.2. 画面構成

Scratch の画面構成は下記の通りです。



※本書で掲載している画面は 2019 年 3 月時点のものです。現在の画面構成と異なる場合があるため、ご了承ください。


2.3. 操作方法

「ブロックパレット」には、Scratch の動作を制御する「ブロック」と呼ばれる部品が多数用意されています。「ブロック」を「スクリプトエリア」に配置し、組み合わせることで、プログラムを作成できます。

画面右側の「ステージ」には、さきほど登場した猫のキャラクターのような「スプライト」を表示できます。スプライトとは簡単に言えば、**画面に登場するキャラクター**のことです。画面右下の「スプライトリスト」で、新しいスプライトを追加できます。

上部メニュー「チュートリアル」からは、Scratch の使い方を説明するヘルプを表示できます。

2.3.1 スプライト

「スプライトリスト」の「新しいスプライト」の右下にある （スプライトを選ぶ）をクリックすると、スプライトライブラリーが表示されます。



この画面でスプライトを選択すると、新しいスプライトを追加できます。



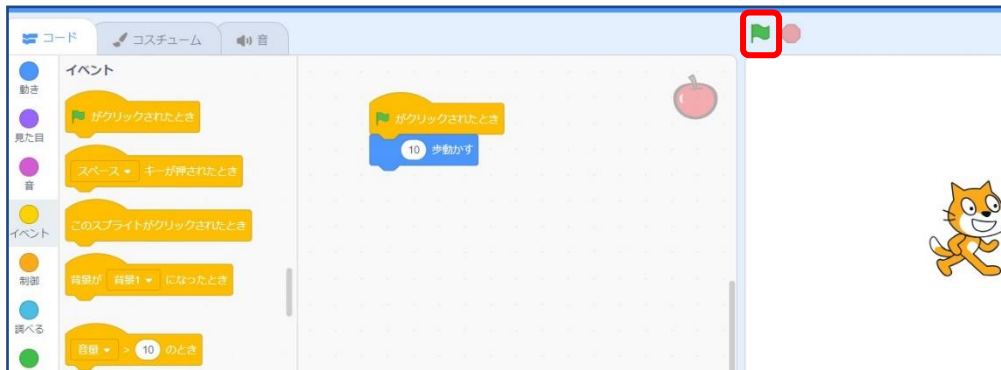
スプライトを切り替えると、「スクリプトエリア」の表示も切り替わり、スプライトごとにスクリプト（プログラム）を作成できます。

2.3.2 プログラムの実行と停止

手順1：「イベント」タブにある「緑の旗がクリックされたとき」ブロックを先頭に置き、続けて実行したいブロックを繋げます。



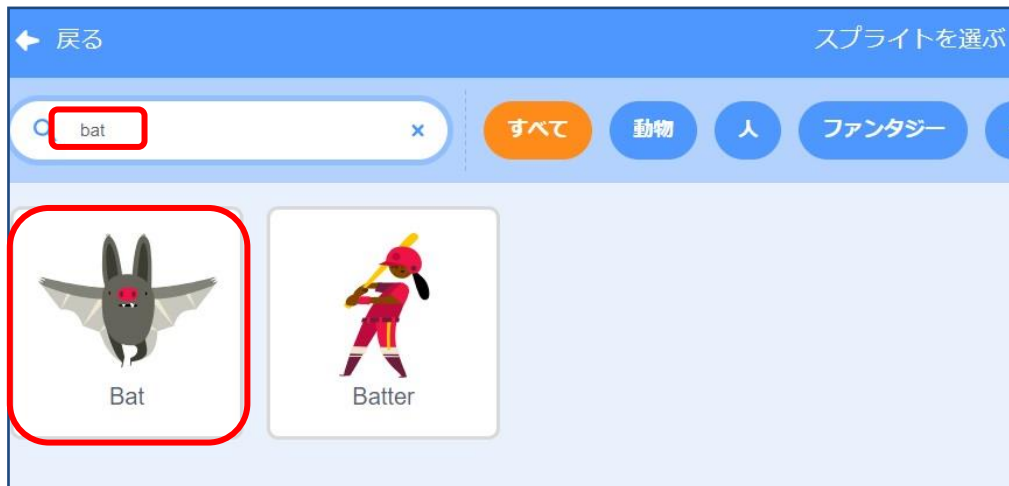
手順2:「ステージ」上部にある緑の旗を押してください。すると、プログラムがスタートします。



緑の旗の隣にある赤いボタンは停止ボタンです。このボタンを押すとプログラムが停止します。プログラムが想定外の動きをしたときは、落ち着いてこのボタンを押しましょう。

演習

先ほどの手順に従い、「bat(こうもり)」の新しいスプライトを登録しましょう。



手順3:「～と言う」ブロックを使用すると、スプライトに発言させられます（吹き出しにメッセージを表示する）。

演習

前の演習で使用したブロックの「こんにちは!」を編集します。

「おはよう!」と発言させてみましょう



2.3.3 コスチューム

「コスチューム」タブをクリックすると、スプライトの「コスチューム」を確認できます。スプライトは複数の「コスチューム」を持ち、切り替えることで、見た目を変化させられます。最初にステージに配置された「スクラッチキャット」には、「コスチューム 1」と「コスチューム 2」の 2 枚のコスチュームが用意されています。

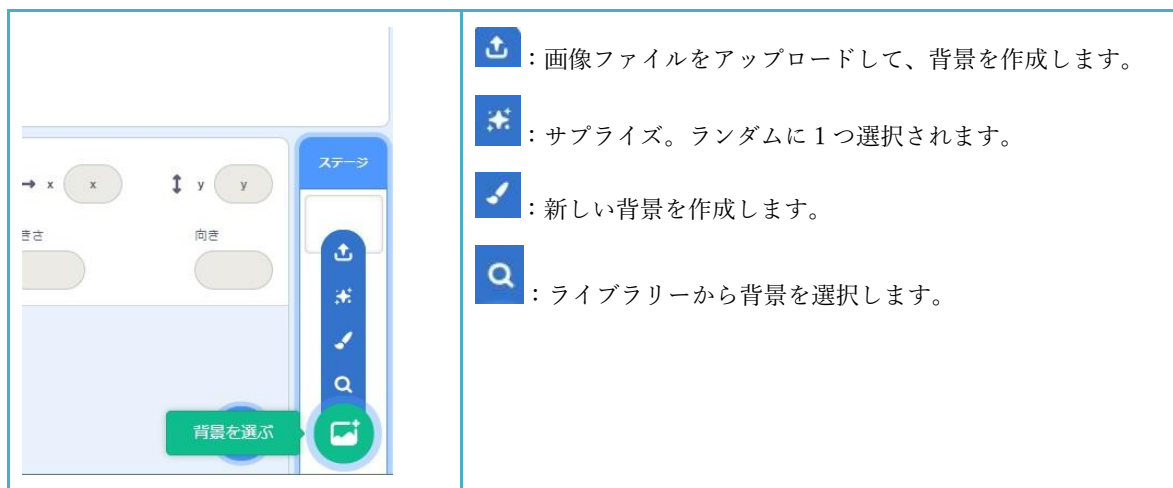
ブロックパレットの、「見た目」の「次のコスチュームにする」ブロックを使用することで、コスチュームを 1 から 2 へ（2 から 1 へ）切り替えられます。また、「コスチュームを costume1 にする」ブロックを使用すると、スプライトのコスチュームを指定したものに切り替えられます。

演習

猫のコスチュームは「コスチューム 1」と「コスチューム 2」が登録されています。「次のコスチュームにする」ブロックをクリックして、コスチュームを切り替えてみましょう。

2.3.4 ステージ（背景）

画面右下の端にある「ステージ」では、背景に関する設定を行えます。



また、ステージの選択中は、（特定のスプライトではなく）ステージに対するスクリプト（ブロック）を設定できます。

演習

新しい背景を作成してください。

演習

「次の背景」ブロックをクリックして、背景を切り替えてみましょう。

3. 「動き」に関するブロック

ここから本格的に各ブロックの機能や性質について紹介していきます。「動き」ブロックは、スプライトの動きを指定して移動させたり回転させたりするブロックです。

	<p>スプライトを、現在向いている方向へ10歩（10ピクセル）動かします。</p>
 	<p>スプライトを左右に回転させます。</p> <p>現在の向き（回転方向）を確認したい場合は、スプライトリストの上部に表示されている詳細情報から確認ができます。</p>
	<p>指定した場所へ移動します。</p> <p>マウスのポインターであればその時のマウスの位置へ。どこかの場所であればランダムな位置になります。別のスプライトがあれば、その位置に移動することもできます。</p>
	<p>座標で指定した位置へ移動します。</p>
 	<p>「〇秒」と指定された時間をかけて、指定した場所へ移動します。ゆっくり移動する、早く移動するという演出が可能になります。</p>
	<p>指定した向き、もしくは物（マウスポインター、スプライト等）のある向きに、スプライトを回転させます。</p>

	<p>x か y の座標を指定した数値ずつ移動させるか、一気に指定した座標へ移動させます。「○歩動かす」との違いは、「向き」に依存しない点です。</p>
	<p>キャラクターが画面の端にぶつかった場合に、跳ね返るように向きを変えます。</p>
	<p>回転方法（スプライトの表示方法）を変えます。「左右のみ」「回転しない」「自由に回転」の3つから選ぶことができます。「左右のみ」にすると、「跳ね返る」動作をしたときなどに、表示が水平方向に反転されます。</p>
	<p>スプライトの現在の x 座標、y 座標、向きを取得します。</p>
	<p>x 座標などの左側のチェックボックスをクリックすると、ステージ上に変数の値を表示させられます。</p>

例題

ブロックを下記のように組み立てて、クリックして動かしてください。
スプライトが左右に動き、画面の端にぶつくと向きを変えます。



4. 「見た目」に関するブロック

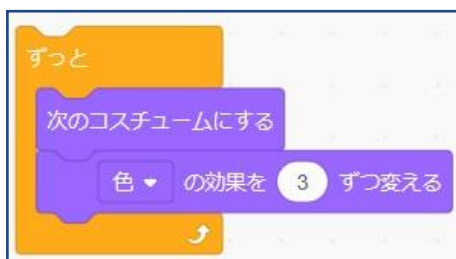
「見た目」ブロックはスプライトの姿を変えたり、スプライトに喋らせたりするためのブロックです。

	<p>吹き出しを表示させます。</p>  <p>※吹き出しの種類が違っただけで同じ効果です。</p>
	<p>コスチュームを指定したもの（次のもの）に切り替えます。</p>
	<p>背景を指定したもの（次のもの）に変更します。</p>
	<p>大きさを変えます。</p>
	<p>画像の色などを変更します。</p> <p>色：色が変化します。 魚眼レンズ：魚眼レンズ状に変化します。 渦巻き：渦巻き状に変化します。 ピクセル化：ぼかしをかけます。 モザイク：画像を小さく、繰り返し表示します。 明るさ：明るさが変化します。 幽霊：色が薄くなります。</p>

	現在設定されている画像効果をすべてリセットします。
	スプライトの表示状態／非表示状態を切り替えます。隠してもあくまで姿が消えるだけで、その場から削除されてしまうわけではありません。
	他のスプライトと重なったときに、最前面や最背面に表示するようにします。
	他のスプライトと重なったときに、1層前面や1層背面に表示するようにします。
	現在のコスチュームの名前や番号を取得します。
	現在の背景の番号や名前を取得します。
	対象のスプライトの現在の大きさを取得します。

例題

ブロックを下記のように組み立てて、クリックして動かしてください。スプライトのコスチュームが切り替わり、猫が歩いているようにアニメーションします。また、色が様々に変化します。

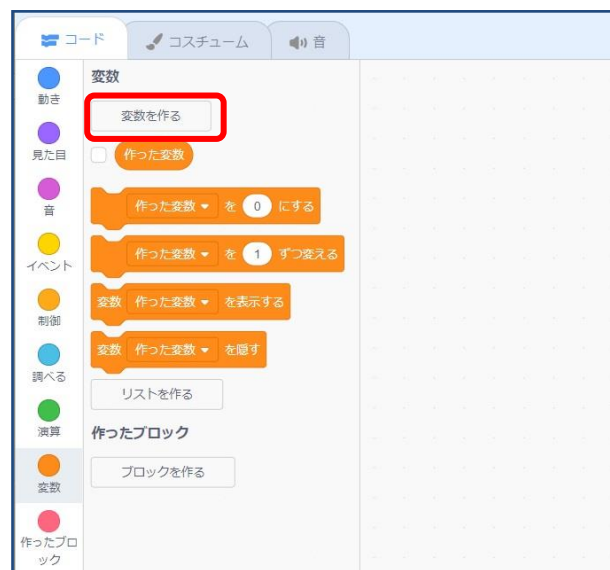


5. 「変数」に関するブロック

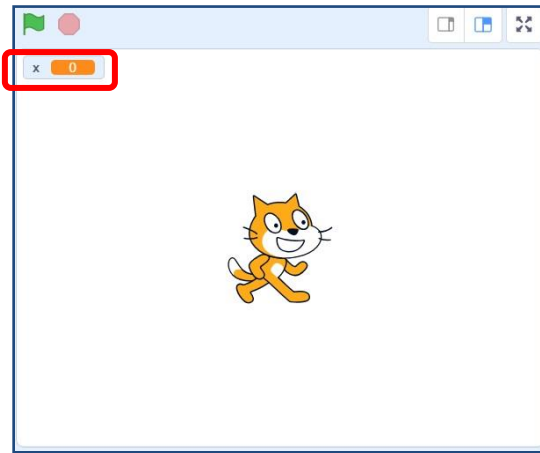
変数とは、「数値」や「文字列」などの何らかの**データを保存するための入れ物**のことです。大半のプログラミング言語には、この変数があります。

変数を使用すると、例えばゲームの「現在の得点」といったデータを一時的に保存でき、ゲーム終了までに何点取れたかを確認できます。

変数を使用するには、ブロックパレットにある「変数」の「変数を作る」ボタンをクリックします。



変数名に「x」と入力して「OK」ボタンをクリックすると、変数「x」が作成されます。その後、ステージに変数「x」が表示されます。



作成された直後の変数には「0」というデータが保存されています。この変数に保存されたデータのことを「値」といいます。また、変数にデータを入れることを「代入」といいます



左のブロックは、変数「x」に値「1」を代入していることを表しています。






演習

変数「score」を作成し、100 を代入してください。

以下は、変数の作成に関するブロックになります。


<p>変数を作る</p>  <p> : 通常の表示</p> <p> : 大きな表示</p> <p> : スライダー</p>	<p>新しい変数を作成します。</p> <p>すべてのスプライト用：すべてのスプライトからアクセスできる変数を作成します。</p> <p>このスプライトのみ：現在のスプライトからのみアクセスできる変数を作成します。</p> <p>ステージに表示された変数をダブルクリックすると、表示状態を切り替えられます。</p> <p>「スライダー」では、スライダー（変数の下部にあるつまみ）を左右に動かすことで、変数の値を変更できます。</p>
<p>リストを作る</p>  	<p>新しいリストを作成します。</p> <p>リストとは、複数の値を扱うためのもので、複数の変数をひとまとめにしたものです。1つ1つの変数には番号が付与されています。</p> <p>すべてのスプライト用：すべてのスプライトからアクセスできるリストを作成します。</p> <p>このスプライトのみ：現在のスプライトからのみアクセスできるリストを作成します。</p> <p>リストの作成直後は、リストの表示は「空」となっています。左下の「+」ボタンをクリックするか、 などのブロックを使用して、値をリストに追加できます。</p>

以下は、変数の操作に関するブロックになります。

	変数の値を取得します。
 	変数の値を変更します。
 	ステージ上での変数の値の表示・非表示を変更します。

例題

変数を作って、その値が0であることを確認してください。


次に  ブロックを作り、クリックしてください。クリックするたびに、値が1増えていくのを確認してください。

以下は、リストの操作に関するブロックになります。

	リストの（すべての）値を取得します。
	リストの末尾に指定された値を追加します。
	指定した位置の値をリストから削除します。
	リストの指定した位置に値を追加します。
	リストの指定した位置の値を別の値に書き換えします。
	リストの指定した位置の値を取得します。

	リストの長さ（入っている値の数）を取得します。
	リストに、指定した値が含まれているかどうかを調べます。
 	ステージ上でのリストの値の表示・非表示を変更します。

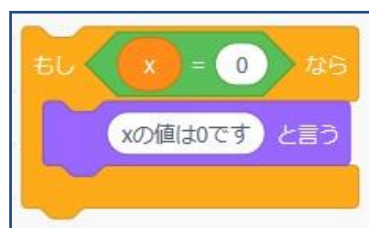
例題

リストを作り、「hello」と「world」を追加してください（リスト左下の「+」を使用します）。「Hello！と言う」ブロックにリストを組み合わせて、（リスト＋「〇〇と言う」）ブロックを作り、クリックして実行してください。リストに追加された文字列から、吹き出しが作成されます。



6. 「制御」に関するブロック

「制御」というブロックパレットにある「もし〇〇なら」のようなブロックを使用すると、**ある条件に応じた処理**を行えます。条件によって、動きが分岐していくことを「**条件分岐**」と言います。たとえば、変数「x」の値に応じて、メッセージを表示するという処理は、次のようになります。



※上記の図にある緑色のブロックは「演算」にあります。「演算」ブロックに関しては次で紹介します。

x の値が 0 の状態で、このブロックをクリックして実行します。すると、結果は以下のようになります。



条件が成立した場合と、そうではない場合で、処理を分岐させることもできます。次の「もし〇〇なら、でなければ」ブロックを使用します。



上記のブロックをクリックして実行した結果は次のようになります。



演習

変数「x」の値に応じてメッセージを表示する処理を実装してください。

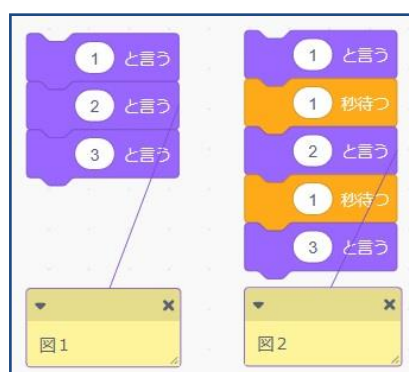
「制御」ブロックは、条件に応じて処理を変えたり、同じ処理を繰り返すなどの制御を行うためのブロックです。使いこなせば複雑なプログラムが書けるようになります。

	指定された秒数、処理をストップさせます。
	指定された回数、間に挟まれたブロックの処理を繰り返します。
	無条件に、間に挟まれたブロックの処理を繰り返します。
	指定された条件が成立した場合は、間に挟まれたブロックの処理を実行します。
	指定された条件が成立した場合は、1 つめの隙間に挟まれたブロックの処理を実行します。そうでない場合は、2 つ目の隙間に挟まれたブロックの処理を実行します。

	指定された条件が成立するまで、処理をストップさせます。
	指定した条件が成立するまで、処理を繰り返し実行します。
	スクリプトの実行を停止させます。
	クローンされたときに呼び出されます。
	自分自身（または指定したスプライト）のクローン（複製）を作成します。
	クローンを削除します。

例題

ブロックを「図1」のように組み立てて、クリックして動かしてください。処理が連続して実行されるため、最後の「3 と言う」だけが吹き出しに残ります。次に「図2」のように、「1 秒待つ」ブロックを挟んで、再度実行してください。今度は待ち時間が入るため、「1 と言う」「2 と言う」「3 と言う」の吹き出しがそれぞれ確認できます。



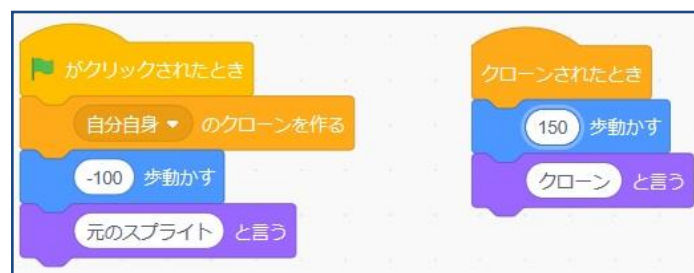
例題

ブロックを下記のように組み立てて、クリックして動かしてください。スプライトが徐々に大きくなります。(大きさを 100%にする、で元に戻せます)









例題



ブロックを下記のように組み立てて、緑の旗をクリックして動かしてください。スプライトのクローンが作成されます。元のスプライトは左へ動き、クローンのスプライトは右に動きます。



7. 「演算」に関するブロック

「演算」ブロックは数学的な計算や、等号や不等号による比較を行ったり、文字列を操作したりするブロックです。

	+ (加算)、- (減算)、* (乗算)、/ (除算) の演算を行います。
	指定された範囲の乱数を生成します。
	< (より小さい)、= (等しい)、> (より大きい) の比較演算を行います。
	2 つの条件が両方とも成立するかどうかを調べます。
	2 つの条件のうち少なくとも 1 つが成立するかを調べます。
	指定された条件が成立しないかどうかを調べます (true/false を反転させます)
	2 つの文字列を結合します。
	文字列から指定された位置の文字を取り出します。
	文字列の長さ (文字数) を調べます。
	割り算の余りを返します。

	<p>数値を四捨五入します。</p> <p>例：3.4 → 3</p> <p>例：3.5 → 4</p>
	<p>指定された数値の絶対値などを求めます。</p>

演習

加算のブロックを使用して、足し算の結果を表示してください。

ただし、2つの数はそれぞれ変数に保存したものを使用してください。




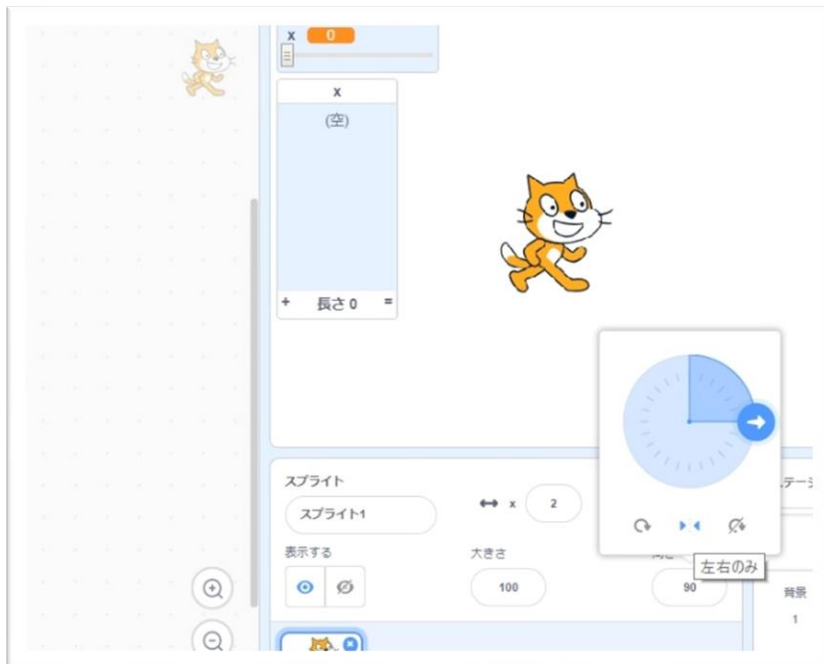
8. 「イベント」に関するブロック

「イベント」とは、プログラムの世界では「きっかけ」「出来事」といった意味を持ちます。「イベント」ブロックは、何をきっかけにそのプログラムが動作するかを決めるためのブロックです。

	緑の旗（ステージ左上）がクリックされたときに呼び出されます。
	指定したキーが押されたときに呼び出されます。
	スプライトがクリックされたときに呼び出されます。
	背景が指定のものになったときに呼び出されます。
	音量（タイマー、ビデオモーション）が指定された数値以上になったときに呼び出されます。 音量：マイクの音量 タイマー：タイマーの値 ビデオモーション：カメラの動き
	指定のメッセージを受け取ったときに呼び出されます。
	指定のメッセージを送ります。続くブロックがあれば、すぐに実行されます。 「新しいメッセージ…」を選択すると、任意のメッセージを登録できます。
	指定のメッセージを送ります。続くブロックがあれば、メッセージ送信先の処理が終了してから、実行されます。

例題

スプライトの「回転の種類」を  (左右のみ) に設定してください。スプライトの詳細の「角度」から変更ができます。次に、ブロックを下記のように組み立ててください。左右キーを押すと、スプライトが左右に向きを変えます。



次に「メッセージ」という機能について確認しましょう。

メッセージはスプライトから別のスプライトに送るサインのようなもので、それをきっかけにして動き出す、というプログラムが作成できます。「メッセージを送る」、ではそのまま送った側のプログラムも継続して動きますが、「メッセージを送って待つ」では送った側は受け取った側の動作が終わるまで動きを中断するという違いがあります。

例題

猫のスプライトに、①のブロックを配置しましょう。次に、新しいスプライトを作り、②のブロックを配置しましょう。猫のスプライトのブロックをクリックすると、新しいスプライトで「こんにちは!」という吹き出しが2秒間表示され、その後、猫のスプライトに「やあ!」という吹き出しが表示されます。

①



②

