

「やってみよう プログラミング」 補助資料 -Scratch 編-

Scratch は MIT メディア・ラボのライフロング・キンダーガルテン・グループによって開発されました。
詳しくは <http://scratch.mit.edu> をご参照ください。

● p5 スプライトをダンスさせるスクリプト



● p5 Challenge 「新しい音を読み込んだり、録音したり、新しいコスチュームにカメラを使ってコマ撮りしたりすることで、マルチメディアの学習を考えてみましょう」

(実践のヒント)

- ・新しい音を読み込みたいとき…スクリプトエリア上部のタブ「音」を選択することで、音声の読み込みや録音ができます（読み込む場合は音声を用意しておく必要があります）。
- ・新しいコスチュームを追加したいとき…コスチュームを追加したいスプライトを、スプライトエリアから選択した後、スクリプトエリア上部のタブ「コスチューム」を選択することで、コスチュームの追加ができます。

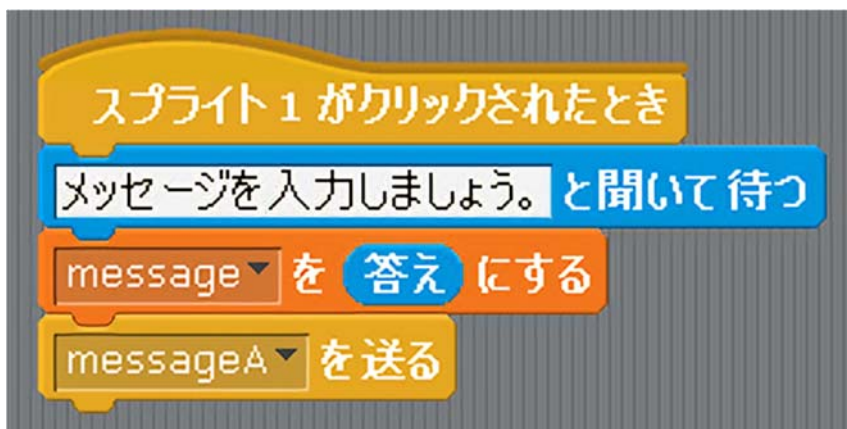
● p6 ネコ（スプライト1）をクリックすると、クリックされた総回数をネコ（スプライト1）がセリフとして言い、スペースキーを押すと回数がリセットされて0に戻るスクリプト



- p6 Challenge 「クリックされた回数が偶数の時にネコ（スプライト1）が鳴く」 スクリプト例
※スペースキーを押すと回数がリセットされるスクリプトは変更がないため割愛しています。



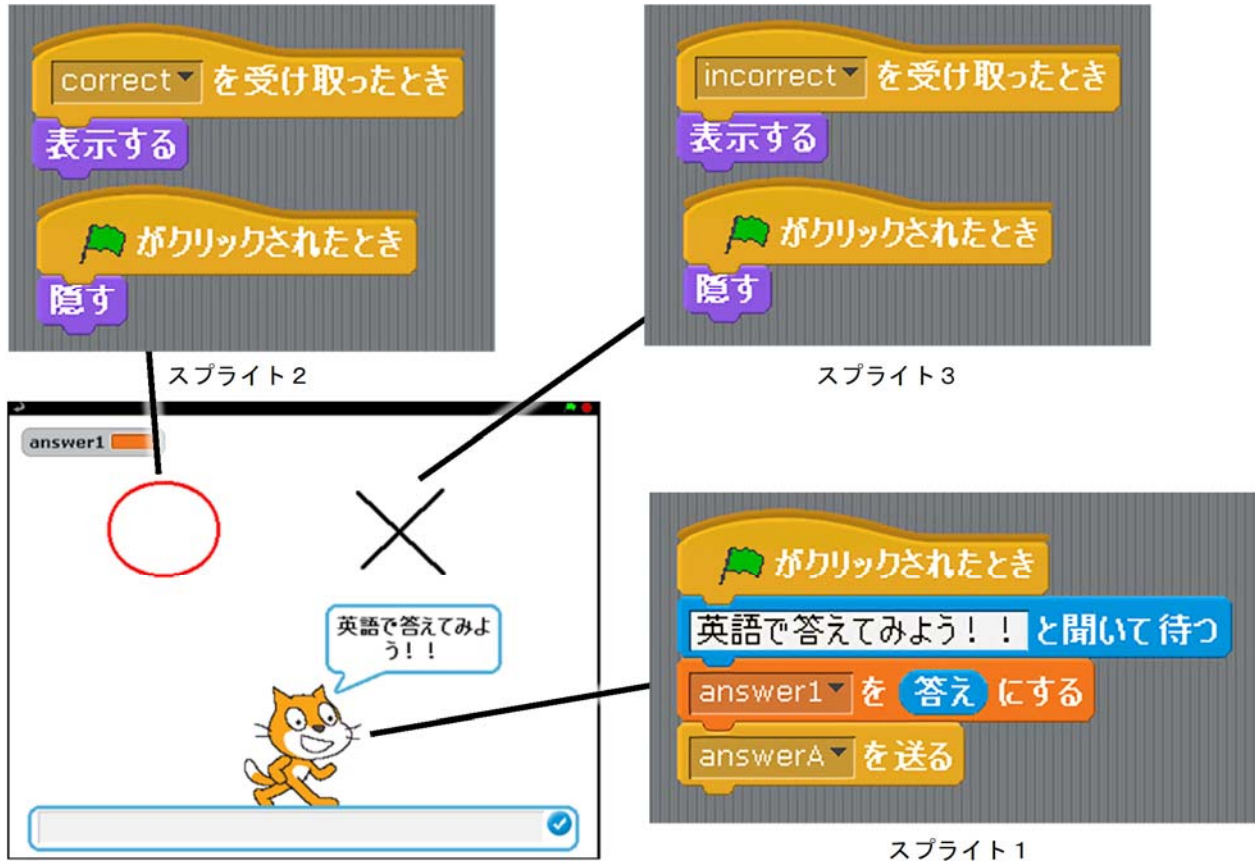
- p8 ① ネコ（スプライト1）をクリックすることで任意の文字列を入力し、「messageA」を合図に文字列を送るスクリプト（Aさん側）



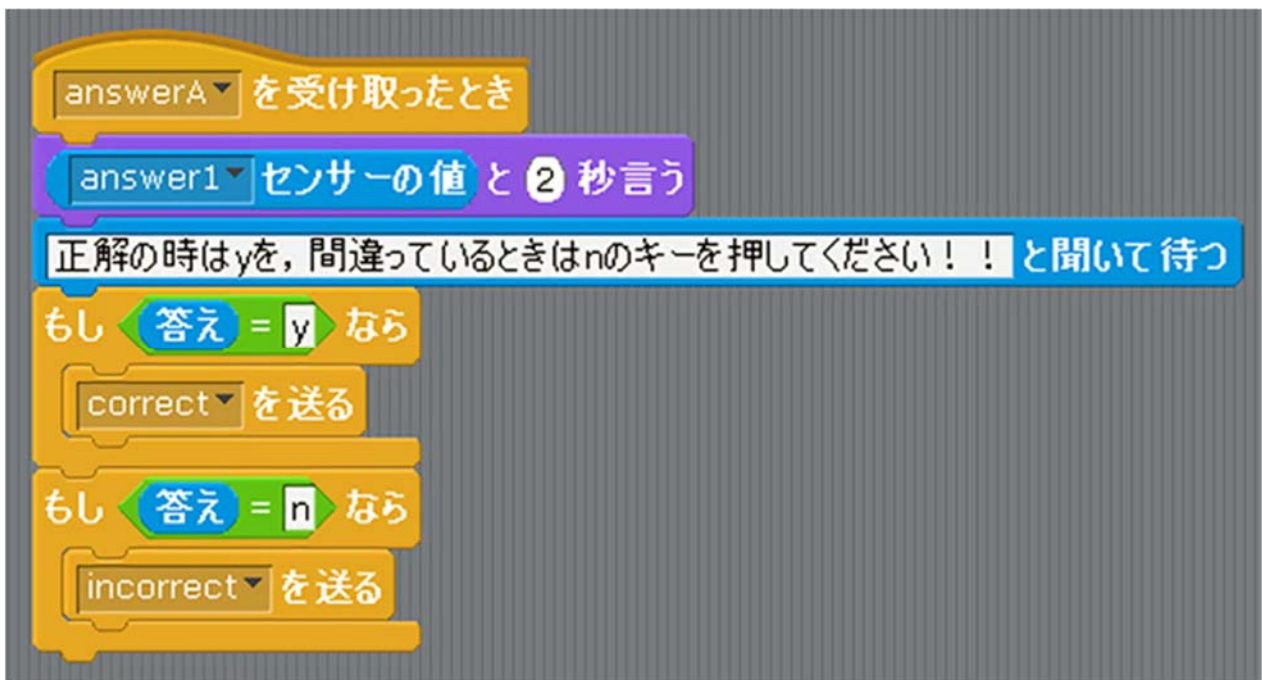
- p8 ② 「messageA」の合図を受け取って共有された変数（message）の値を表示するスクリプト（Bさん側）



● p9 ① Aさんの画像イメージとプログラム

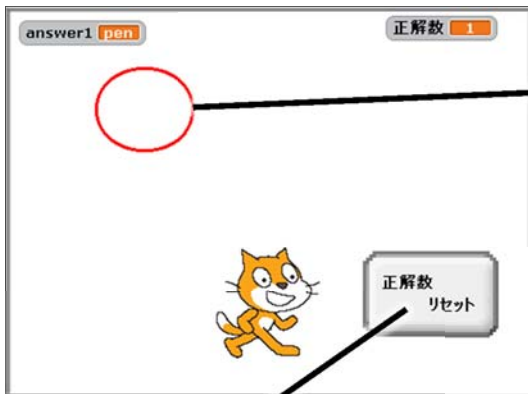


● p10 ③, ④ Bさんのスクリプト (Aさんが入力した文字列 (answer1) を2秒間表示し, 正解であれば「y」, 不正解であれば「n」の入力を促す)



● p10 Challenge 「繰り返しスクリプトを実行することを前提に、正解した数をカウントして表示する等の工夫をしてみよう」

(考え方) 問題に正解すると右上の変数「正解数」が1つずつプラスされていきます。



あらかじめ、変数「正解数」を用意します(変数の使い方については p.6 を参照)。それから、正解したときに表示する「○ (スプライト2)」に、「correct」を受け取ったときに「表示する」プログラムの下に「正解数を1ずつ変える」というブロックをつけ足します。



正解数をリセットするためのブロック(スプライト4)を作ります。クリックされたときに、変数「正解数」の値を0に戻します。

● p11 ① Aさんの画面イメージとプログラム



● p11 ② Bさんの画面イメージとプログラム。



● p12 Challenge 「会話の履歴が残るにはどうしたらよいか？」

(考え方)

- ・変数ではなく、リストを使用します。
- ・自分と相手の発言内容を、送受信時にリストに加えていくようにします（例えば、「(なにか) を (リスト) に追加する」のブロックを使用します)。

● p12 Challenge 「メッセージが送られてきたことを相手に確実に通知するにはどうしたらよいか？」

(考え方)

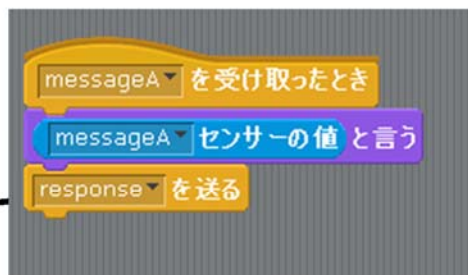
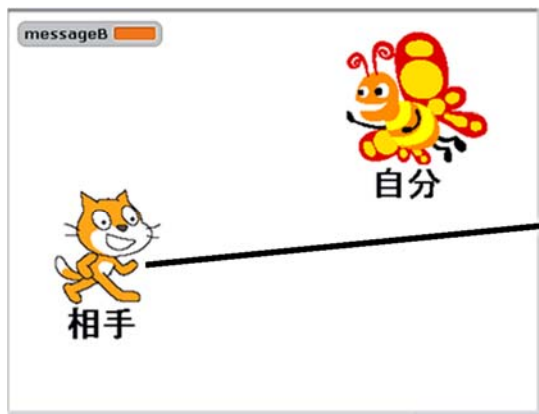
メッセージが伝わったことを知らせる方法としてさまざまなものが考えられますが、視覚的に知らせる方法を例示します。送信が完了したら送信側の画面の「相手」とかかれたspriteが点滅して、あたかも相手が受信しているかのようなプログラムを制作します。

・Aさん（送信側）のスク립トエリアの「コスチューム」タブ



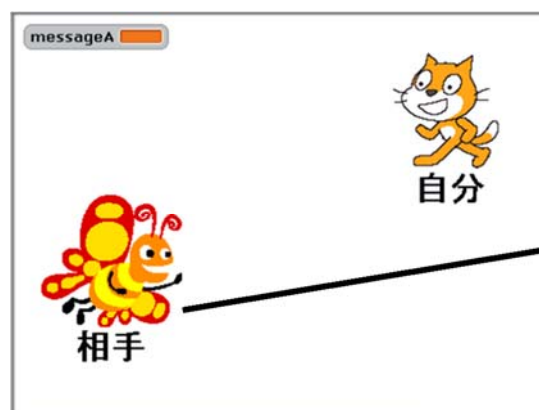
spriteの点滅はコスチュームの切り替えで表現するため、色を変えた2つめのコスチュームを用意します（任意で設定します）。

・ B さん（受信側）の画面イメージとプログラム



受信側の「相手」（ここではネコ（cat））に「response を送る」というブロックを追加します。

・ A さん（送信側）の画面イメージとプログラム



送信側の「相手」（ここでは蝶（butterfly））にプログラムを追加します。受信側に届いたことが「response」ブロックを通じて返ってきたら、スプライトを点滅させます（ここでは0.1秒刻みで10回、コスチュームを変化させることで点滅を表現しています）。

● p12 Challenge 「p9~10のクイズ作品と組み合わせて、お互いに問題を出し合い、結果を返す作品を作ってみよう！」

（考え方）

- ・ 双方向性のチャットに、p.9の「○（スプライト2）」と「×（スプライト3）」を追加します。
- ・ 「自分」のスプライトに、p.10の④のような、正解、不正解を送るスクリプトを追加します。

● p12 「安全性を高めるために使用時にパスワードを入れる」

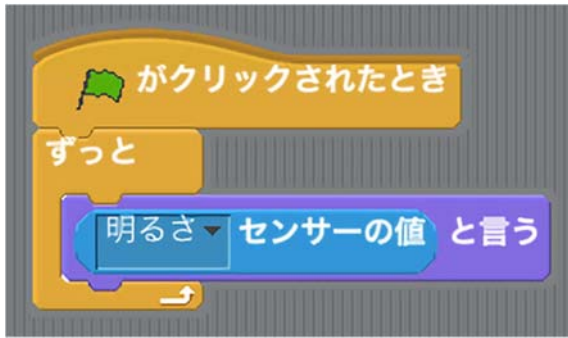
画面イメージとプログラム

The image displays a Scratch chat interface and its associated code blocks. The interface shows a 'messageA' window with a 'パスワード設定' button (1), a cat sprite labeled '自分' (2), and a bee sprite labeled '相手' (3). The code for the '自分' sprite includes: 1. 'Sprite 1 がクリックされたとき' (Sprite 1 clicked) - 'パスワードを設定してください。と聞いて待つ' (Ask for password) - 'パスワード' dropdown - 'を 答え にする' (Set answer). 2. 'cat がクリックされたとき' (cat clicked) - 'パスワードを入力してください。と聞いて待つ' (Ask for password) - 'もし 答え = パスワード なら' (If answer equals password) - 'メッセージを入力してください。と聞いて待つ' (Ask for message) - 'messageA' dropdown - 'を 答え にする' (Set answer) - '答え と言う' (Say answer) - 'messageA' dropdown - 'を送る' (Send message) - 'でなければ' (Otherwise) - 'パスワードが違います。もう一度クリックしてパスワードを入力してください。と言う' (Say password is wrong).

(手順)

- ①パスワード設定ボタン (スプライト 1) を作成します。ボタンを押すと、「パスワード設定」を求められます。
 - ②「自分」のスプライト (cat) をクリックすると、パスワードの入力を求められます。
 - ③設定したパスワードと認証を行い、一致すればチャットを開始します。間違えらるともう一度入力を求められます。
 - ④B さんの方も同様に作成することで、お互いにパスワードを設定し、解除してからチャットを行うプログラムとなります。
- ※注意：ここでは、パスワードの設定から解除の一連の流れを表現するため何度でもパスワード設定ができるようになっています。現実のプログラムでは、最初の 1 回、もしくは変更のためにはそれなりの手続きが必要になることも忘れずに生徒に伝えたいところです。

- p15 ⑤ センサーの値をネコ（スプライト1）に表示させる



- p16 ② センサーで測定した明るさによってネコ（スプライト1）の言葉が変わる



● p16 Challenge 「微妙な明るさのときでも表現が安定するようにしてみよう！」

(考え方)

この Challenge は、生徒が計測・制御を行うときに、センサーの値によって変化することが正解と見なすことから設定しました。実際に計測・制御を行う機器、例えばエアコンでは、設定温度で ON/OFF の制御はしていないと思われます。実際の生活とのつながりを意識したものとなるように、次のスクリプト例を示します。

スクリプト例

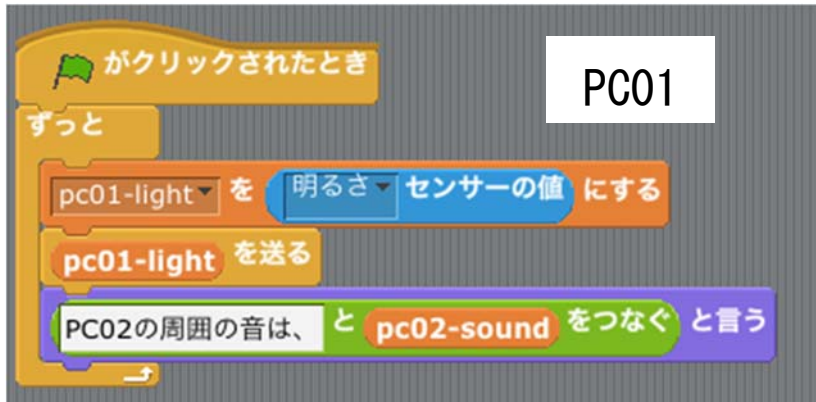
The image shows a Scratch script for a challenge. It starts with a 'when clicked' event block. The first block is a 'say' block: '今の明るさは、' followed by a 'brightness sensor value' block and 'をつなぐ' and 'と言う'. This is followed by a 'loop' block labeled 'ずっと' (forever). Inside the loop, there are several blocks: 'delete all items from buf-list', 'insert into buf-list at index 1 the brightness sensor value', 'wait 0.2 seconds', 'insert into buf-list at index 2 the brightness sensor value', and an 'if' block: 'if (buf-list at index 1 < buf-list at index 2 + 10) then'. Inside this 'if' block, there is a 'say' block: '今の明るさは、' followed by a 'brightness sensor value' block and 'をつなぐ' and '言う'. Below the 'if' block is a 'do otherwise' block. Inside it, there is another 'if' block: 'if (buf-list at index 1 > buf-list at index 2 - 10) then'. Inside this second 'if' block, there is a 'say' block: '今の明るさは、' followed by a 'brightness sensor value' block and 'をつなぐ' and '言う'. Finally, after the 'do otherwise' block, there is a 'say' block: '今の明るさは、' followed by a 'buf-list at index 1' block and 'をつなぐ' and '言う'. The script ends with a 'return to top' arrow.

- ・このスクリプト例では、明るさセンサーの値が直前と比較して「10」より大きい差が無いときには無視をするようにしています。
- ・リストという機能を使い、センサーの値をリストの1番目と2番目に入れ、それを比較しています。

● p16 Challenge 「Mesh を利用して、センサーの値を違うコンピュータに送れるようにしてみよう」

※Mesh 機能を ON にして、変数（ここでは「light」と「sound」）を共有していることを前提に、以下の例を提示します。コンピュータは 2 台使用してそれぞれ、「PC01」「PC02」とします。

スクリプト例



(考え方)

- 変数にセンサーの値を入れることが鍵となります。
- 変数名は自由に設定できますが、MESH ネットワーク内で同じ変数名を使わないように注意して下さい。同じ変数名を利用すると、ネットワーク内に異なるコンピュータから異なる値が送信されるため、以下の不具合が出る可能性があります。
 - 値の異常
 - 2つのデータが繋がった値
(例)「センサー1 = 2」, 「センサー2 = 1 0」
のとき、「2 1 0」と表示される可能性があります。
- 下側のスクリプト例は表示側です。
数値だけの表示もできますが、「明るさは、」と付け加えることもできます。センサーを接続して送信側で付けることも可能ですが、日本語を変数内に格納して送ることはできません。また、センサーの値の再利用が難しくなるのでおすすめしません。

● p17 ② 複数のセンサーを同時に利用する例

がクリックされたとき

30 < 明るさ センサーの値 > まで繰り返す

雨は降るかな? という

雨ですよ! という

がクリックされたとき

抵抗-A センサーの値 < 40 > まで繰り返す

雨は降るかな? という

雨ですよ! という

● p17 Challenge 「複数のセンサーを使って目的の動作をさせてみよう」

ここでは変数を使用して、複数のセンサーを使って目的の動作をさせています。p. 17 で示されたやり方でもできますが、他で使用しやすくするために整理をしています。

スクリプト例



(考え方)

- 湿度センサーと照度センサーを使用し、両方の値が 50 より大きければ「雨が降りそうです」と表示し、どちらかの値が 50 より大きければ「雨雲が近づいている可能性があります」と表示します。
- 上のスクリプト例では、変数を初期化し、センサーの値を変数に代入しています。
- 下のスクリプト例では、2つのセンサーの値を比較し、条件によって文字の出力を変えています。