



ここまでで、スクラッチの基本的な機能についてはかなり学んできたこととなります。そこで、ここからは、これまで学んだことを応用し、簡単なゲームを作りながら、さらにいろいろな機能を学んでいくことにします。

**【課題 13】 新規ファイルに、風船のキャラクターを追加し、風船を少し小さくして、下から上に動かせ、上に風船が上がったら消えるようにしよう。**

- 1 新しいプログラムを作る。 **ファイル** → **新規**
- 2 新しい風船の sprites を表示する
  - (1) 画面右下の  の部分へマウスを移動した後すぐ上のルーペの部分 **sprites を選ぶ** をクリック。
  - (2) 新しく風船の sprites を追加するため、画面右下の  の部分へマウスを移動した後すぐ上のルーペの部分 **sprites を選ぶ** をクリックし、一覧の中から **Balloon1** をクリックする。
  - (3) 最初にあったネコの sprites は削除する。



**3 風船を少し小さくし(80%程度)、下から上に動かすプログラムを作成する。**

- (1) **見た目** カテゴリの **大きさを 100%にする** を選び、100 を 80 に変更する。
- (2) y 座標の値を一番下 (-170 程度) にする。
- (3) **制御** カテゴリの **ずっと** の中に、**動き** カテゴリの **y 座標を 10 ずつ変える** のコードを入れ、10 の数字を適当に変更する。



ここまでのプログラムで、下から上に上がるようになったが、このままでは、ずっと上がり続けるため、上に上がったら、消えるようにしよう。

**4 「上に上がったら、風船が消える」ようにする**

- (1) 風船が上にあるときの y 座標を調べる。120 とする
- (2) **制御** のカテゴリから、**もし・・・なら** を下につなげる
- (3) **演算** のカテゴリから、をスクリプトエリアに出し、50 を 120 に修正する



- (4) **動き** のカテゴリから、**y 座標** を左の空欄に入れる

(5) **見た目** のカテゴリから **隠す** を選び、**もし・・・なら** の下に入れる



このようにして作った下のプログラムを実行して、動作を確認してみましょう。

予想した動きになったでしょうか。

では、もう一度このプログラムを実行してみてください。

今度の画面は白紙のままになっているはずです。

その理由と、どうしたらよいかを考えるのが、次の練習問題です。



<練習問題 13> 旗をクリックするたびに、風船が下から上に動き、上に行ったら消え、すぐ次の違った色の風船が上がるようにしよう。

【課題 1 4】練習 1 3 を、いろいろな場所から風船が上がるようにしよう。(乱数の利用)

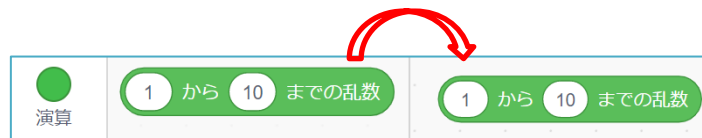
＜考え方＞ x 座標がいろいろな数値になるようにする (乱数)

1 練習 1 3 を表示し、コピーを保存し、ファイル名を課題 1 4 に変更しておく。

練習 13 の完成例



2 演算 のカテゴリーから、1 から 10 までの乱数 を選び、空きスペースへ入れる。



3 数値を 画面の x 座標をいっぱい -200 から 200 までに変える



4 動き カテゴリーの x 座標を〇にする のブロックを空きスペースに入れ、その数値のところにこのブロックを入れる。



5 練習 1 3 のプログラムのどこに入れたらいいか考える。

6 旗をクリックして、いろいろな位置から風船が上がることを確かめる。

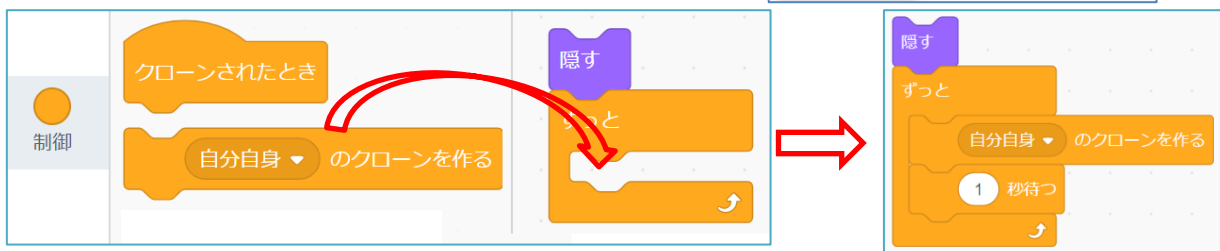


課題 1 4 の完成例

<練習問題 14> 課題 1 4 を風船が 5 個出たら終わり、画面から消えるようにしよう。

**【課題 15】 旗をクリックするといろいろな色の風船がいろいろな位置から続けて出てくるようにしましょう。(クローンの利用)**

- 1 課題 14 を表示し、課題 15 に変更しておく  
(コードは前ページに表示)
- 2 元のコードの旗が押されたときの下のかたまりを移動し、分離する。
- 3 **見た目** のカテゴリから、**隠す** を選び、旗が押されたときの下に置く。
- 4 **制御** のカテゴリから、**ずっと** を選び **隠す** の下に置く。
- 5 **制御** のカテゴリから、**自分自身のクローンを作る** を空きスペースに置く
- 6 **ずっと** の中に入れる。
- 7 **制御** のカテゴリから、**1秒待つ** を下につなげる。



- 8 先ほどの **旗が押されたとき** のブロックをこのプログラムの上に置く。



- 9 **制御** のカテゴリから、**クローンされたとき** を選び、もとのコードの上のせる。



10 右図のようにコードを変更する。

動作確認をしてみると、だいたいうまく  
いっているが、風船の色が変わらない。

```
クローンされたとき
  大きさを 80 %にする
  表示する
  x座標を -200 から 200 までの乱数 、y座標を -170 にする
  ずっと
    y座標を 5 ずつ変える
    もし y座標 > 120 なら
      隠す
```

風船の色を変えるために、次のコードを入れる。

- (1) **演算**カテゴリーの **1から10までの乱数** ブロックを空きスペースに置く。
- (2) 数値を1から3までに変更する。
- (3) **見た目**カテゴリーの**コスチュームを○にする** のブロックを空きスペースに置く。
- (4) ○の部分に乱数のブロックを入れる。
- (5) このブロックをコードの中に入れる。

```
が押されたとき
  隠す
  ずっと
    自分自身 のクローンを作る
    1 秒待つ
  クローンされたとき
    大きさを 80 %にする
    表示する
    コスチュームを 1 から 3 までの乱数 にする
    x座標を -200 から 200 までの乱数 、y座標を -170 にする
    ずっと
      y座標を 5 ずつ変える
      もし y座標 > 120 なら
        隠す
```

課題 15 の完成例

ここではプログラムのかたまりが二つでき、それらが連動して動作することになる。  
旗をクリックして動きを確かめよう。いろいろな色の風船が次々と上がっていくようになっているはずである。

動作確認ができたなら、プログラムの意味を考えてみよう。

- (1) 旗が押される (開始)
- (2) 風船が非表示となる
- (3) 一つ目の自分自身のクローンを作る
- (4) **クローンされたとき** のコードを実行する
- (5) 1秒後に次のクローン風船を表示し、この動作を繰り返す