

KASAI . SOFTWARE LAB



スクラッチ

Scratch

完全攻略マニュアル

Scratch プログラミング 技法を学ぶ

河西朝雄 著

定価 1,500 円 + 税

Scratch 完全攻略マニュアル

Scratch プログラミング技法を学ぶ

河西 朝雄著

カサイ．ソフトウェアラボ

はじめに

与えられた問題をプログラムする場合、いかに理論整然と効率よく、誤りの少ない、良いプログラムを作れるかが重要です。伝統的な学問の数学や物理は方程式などを使って理論的に問題を解決する方法が長年研究されてきました。プログラムの世界は歴史が浅いため理論的な手法が確立していません。

プログラムを使って問題を解くための論理または手順をアルゴリズム (algorithms : 算法) といいます。拙著「Scratch によるはじめてのアルゴリズム入門」では先達が研究した伝統的なアルゴリズムを Scratch で実現する方法を紹介しています。本書で言う「プログラミング技法」とはアルゴリズムほど普遍的ではないが、プログラミングのテクニックや定石的なものを指します。「プログラミング技法」というカードをうまく組み合わせることで効率的なプログラムを作ることができるようになります。

本書では以下の 14 のカテゴリに分けて Scratch でのプログラミング技法を紹介します。

1 章～8 章は Scratch に限らず他のプログラミングにも共通する内容です。9 章～14 章は Scratch ならではの内容です。

- 1 章 制御構造
- 2 章 データ型と演算子
- 3 章 リスト操作
- 4 章 ユーザーブロック
- 5 章 再帰
- 6 章 Scratch の座標
- 7 章 イベント
- 8 章 メッセージ
- 9 章 クローン
- 10 章 調べる
- 11 章 動き
- 12 章 見た目
- 13 章 音
- 14 章 ペン機能

Scratch を用いて高度なプログラミングを勉強したい人にとって、本書で紹介したプログラミング技法を学ぶことが少しでもお役に立てば幸いです。

2018 年 1 月
河西 朝雄

目次

1 章 制御構造	9
1-1 for 文	10
1-2 while 文	13
1-3 do while 文	14
1-4 if else 文	15
1-5 else if 文	18
1-6 switch case 文	20
1-7 break 文	21
1-8 無限ループ	22
1-9 プログラムのスリープと停止	23
2 章 データ型と演算子	25
2-1 扱えるデータ型	26
2-2 数値演算	31
2-3 文字列演算	36
2-4 論理演算	40
2-5 変数の操作	43
2-6 乱数	45
3 章 リスト操作	49
3-1 リスト要素の参照	50
3-2 要素数 N のリストの生成	51
3-3 リスト A→リスト a へコピー	52
3-4 リストへの追加	53
3-5 リストからの削除	54
3-6 隣接項の操作	55
3-7 リストの 2 次元化 (表 : テーブル)	60
3-8 初期化データ	63
3-9 リストを出力画面に使う	65

4 章 ユーザーブロック	69
4-1 ブロックの定義	70
4-2 引数の型	72
4-3 ブロックの戻り値	73
4-4 ローカル変数	74
4-5 リストの引数	76
5 章 再帰	77
5-1 ハノイの塔	78
5-2 再帰と戻り値	80
5-3 戻り値を 2 箇所で使う場合	82
5-4 再帰とローカル変数	84
6 章 Scratch の座標	87
6-1 スプライトの座標	88
6-2 スプライトの角度	90
6-3 マウスの座標	92
6-4 関数のグラフ	95
6-5 ウィンドウとビューポート	97
7 章 イベント	103
7-1 クリックイベント	104
7-2 キーイベント	106
7-3 背景イベント	108
7-4 タイマーイベント	111
8 章 メッセージ	113
8-1 メッセージのやりとり	114
8-2 メッセージ名を変数で管理	116
8-3 開始の通知	118
8-4 判定の通知	120

9 章 クローン	123
9-1 自分自身のクローンを作る	124
9-2 クローン生成時の通知イベント	125
9-3 クローン生成のタイムラグ	128
9-4 自分自身以外のクローンを作る	129
9-5 クリック位置のクローンを別のクローンで置き換える	132
10 章 調べる	135
10-1 触れたかどうか	136
10-2 マウス情報	141
10-3 スプライトの情報	143
10-4 キーが押されたか	145
10-5 日時	146
10-6 データ入力	148
11 章 動き	149
11-1 移動	150
11-2 回転	152
11-3 マウスへの追従	154
11-4 端で跳ね返る	155
12 章 見た目	157
12-1 話す	158
12-2 表示・非表示と階層	159
12-3 コスチューム	162
12-4 画像効果	163
12-5 大きさ	166
12-6 背景	167
12-7 背景のスク립ト	168

1 3 章 音	173
13-1 音の登録	174
13-2 音を鳴らす	176
13-3 リズム楽器を鳴らす	178
13-4 楽器を音符で鳴らす	180
13-5 音符データをリストに格納	182
13-6 音量とテンポ	184
1 4 章 ペン機能	185
14-1 ペン機能を行うスクリプト	186
14-2 スタンプ	189
14-3 ペンの設定ブロック	190
14-4 setpoint と moveto	191
14-5 move	193
14-6 グラフィックス用ブロック	194
14-7 リカーシブグラフィックス	201

1 章 制御構造

Scratch の繰り返し制御構造には **for** 文、**while** 文に相当するものがあります。



条件判定構造には **if else** 文に相当するものがあります。



Scratch には **do while** 文、**else if** 文、**switch case** 文、**break** 文はある制御構造を使って工夫して作らなければなりません。
この章では以下の内容について説明します。

- 1-1 for 文
- 1-2 while 文
- 1-3 do while 文
- 1-4 if else 文
- 1-5 else if 文
- 1-6 switch case 文
- 1-7 break 文
- 1-8 無限ループ
- 1-9 プログラムのスリープと停止

1-1 for 文

for 文を Scratch で書くと「回繰り返す」または「まで繰り返す」で書くことができます。



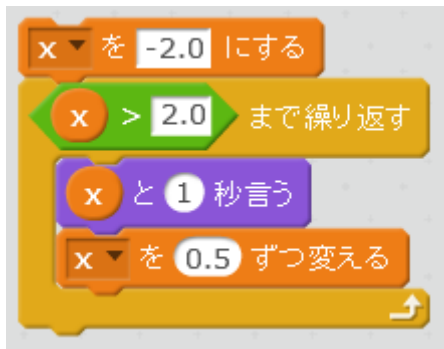
```
for (i=1;i<=10;i++){  
  
}
```

のように繰り返す回数が直感でわかる場合は「回繰り返す」を使います。



```
for (x=-2.0;x<=2.0;x+=0.5){  
  
}
```

のように繰り返す回数が直感的に分かり難いものは「まで繰り返す」を使います。終了条件の「 $x \leq 2.0$ 」は逆の「 $x > 2.0$ 」になります。



■for の二重ループ

```
for (i=1;i<4;i++){
    for (j=1;j<5;j++) {

    }
}
```



という for の二重ループを で書くと以下のようになります。





で書くと以下ようになります。



2 章 データ型と演算子

Scratch で扱えるデータ型は数値型、文字列型、論理型の 3 種類です。

データは変数に格納し、操作します。

規則性がなくまったくでたらめに並ぶ数を乱数といいます。プログラムの世界では乱数を利用することで様々な処理が行えます。

この章では以下の内容について説明します。

2-1 扱えるデータ型

2-2 数値演算

2-3 文字列演算

2-4 論理演算

2-5 変数の操作

2-6 乱数

2-1 扱えるデータ型

Scratch で扱えるデータ型は数値型、文字列型、論理型の 3 種類です。

■データのキー入力

数値型



○の項には数値文字と、マイナス(-)、ピリオド (.)、指数表記(e)しか入力できません。

文字列型



□の項にはすべての文字が入力できます。

論理型



この緑色の項目にはキー入力できません。

1. 数値型

■数値型のデータ範囲

数値型のデータ範囲は整数としての範囲と実数としての範囲があります。

・整数としての範囲

-2147483648～2147483647 (4 バイト int 型相当)

・実数としての範囲

±2.225e-308～±1.797e+308(8 バイト double 型相当)

有効桁数 15～17

整数型の範囲を超えると実数型で扱います。「2147483647+1」は「2147483648.00」として扱われます。

扱える実数の範囲を超えると「Infinity」となります。



■16 進表記

0x1a のような 16 進表記が可能です。



この時点では文字列型としてと数値型の両面を持っています。



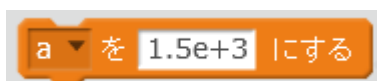
と文字列演算を行えば、「0x1a1」のように文字列として連結されます。



と数値演算を行えば「26」という数値になります。

■指数表記

1.5e+3(1.5×10³)のような指数表記が可能です。



この時点では文字列型としてと数値型の両面を持っています。

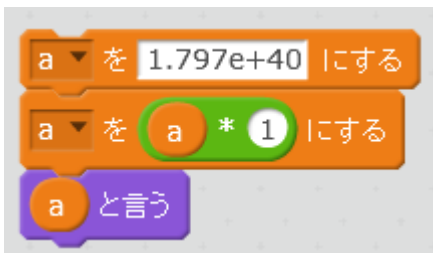


と文字列演算を行えば、「1.5e+31」のように文字列として連結されます。



と数値演算を行えば「1500」という数値になります。

■ 誤差



「17970000000000000083」のように 18 桁目で誤差が出ます。

変数の表示領域は狭いので「17970000000000000083」という表記でなく「1.797e+40」という表記になります。



■数値として表示するのか文字列として表示するのか

変数 **x** に「3.14159」を代入した時点では文字列として扱われます。「~と言う」スクリプトで表示するとそのまま文字列として「3.14159」と表示されます。



ところが「 $x=x*1$ 」という数値演算を行うと、変数 x の値は数値として扱われます。数値データを「~と言う」スクリプトで表示すると、「3.14」のように小数点以下2桁にまるめられて表示されます。



ただし、「0.00123」のような小数点以下2桁目までが「0」の数値は、まるめずにそのまま「0.00123」と表示されます。

2. 文字列型

文字列の長さは 10240 文字までです。以下のプログラムで検証してみましょう。



10241 回「a」を連結しても「10240」文字で打ち切られていることがわかります。

3. 論理型

■論理型の値

論理型は true(真)と false (偽) の真偽値です。



true

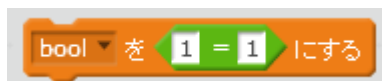


false



■真偽値の変数と引数

変数には真偽値を格納できますが、変数の値を直接、真偽値として扱うことはできません。



たとえば、真偽値の引数 bool に変数 bool を指定することはできません。



3 章 リスト操作

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。一般のプログラミング言語では配列と呼んでいます。この章ではリストの作り方とリストの操作方法について以下の内容を説明します。

- 3-1 リスト要素の参照
- 3-2 要素数 N のリストの生成
- 3-3 リスト A → リスト a へコピー
- 3-4 リストへの追加
- 3-5 リストからの削除
- 3-6 隣接項の操作
- 3-7 リストの2次元化（表：テーブル）
- 3-8 初期化データ
- 3-9 リストを出力画面に使う

3-1 リスト要素の参照

リスト **a** の **i** 番目の要素は **i 番目(a)** で取得できます。

- ・リスト **a** の先頭の要素から最後の要素まで順次表示します。



4 章 ユーザーブロック

予め与えられているスクリプトブロック以外に自分で新しい機能を持つユーザーブロックを作る（定義）ことができます。自分で定義したユーザー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。**Scratch** のユーザーブロックでは戻り値やローカル変数を使用することができません。

この章では以下の内容について説明します。

4-1 ブロックの定義

4-2 引数の型

4-3 ブロックの戻り値

4-4 ローカル変数

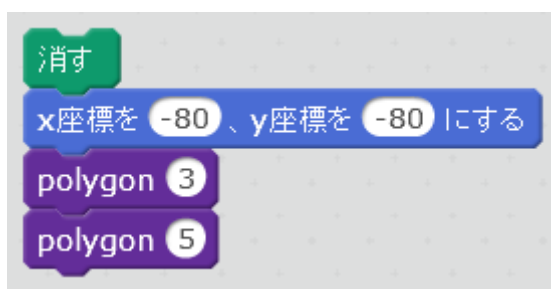
4-5 リストの引数

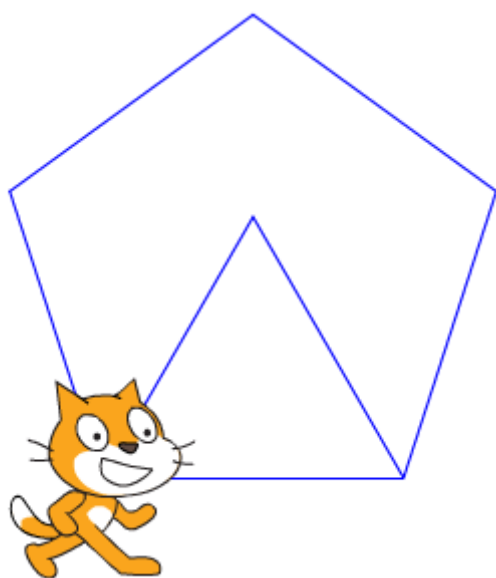
4-1 ブロックの定義

予め与えられているスクリプトブロック以外に自分で新しい機能を持つユーザーブロックを作る（定義）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。

■多角形を描くブロック `polygon(n)`

引数 `n` で与えた角数の多角形を描くユーザーブロックです。





5 章 再帰

再帰という考え方は人間の一般的な感覚からは縁遠いものですがプログラムの世界では重要な考え方です。

再帰的（リカーシブ）な構造とは、自分自身（ n 次）を定義するのに、自分自身より 1 次低い部分集合（ $n-1$ 次）を用い、さらにその部分集合は、より低次の部分集合を用いて定義するということを繰り返す構造です。このような構造を一般に再帰と呼んでいます。

ある手続きの内部で、再び自分自身を呼び出すような構造の手続きを再帰的手続きと呼び、手続き内部で再び 1 次低い自分自身を呼び出すことを再帰呼び出し（リカーシブ・コール）と呼びます。

この章では以下の内容について説明します。

5-1 ハノイの塔

5-2 再帰と戻り値

5-3 戻り値を 2 箇所で使う場合

5-4 再帰とローカル変数

5-1 ハノイの塔

ハノイの塔とは次のようなパズルゲームです。

「3本の棒 a、b、c がある。棒 a に、中央に穴の空いた n 枚の円盤が大きい順に積まれている。これを1枚ずつ移動させて棒 b に移す。ただし、移動の途中で円盤の大小が逆に積まれてはならない。また、棒 c は作業用に使用するものとする。」

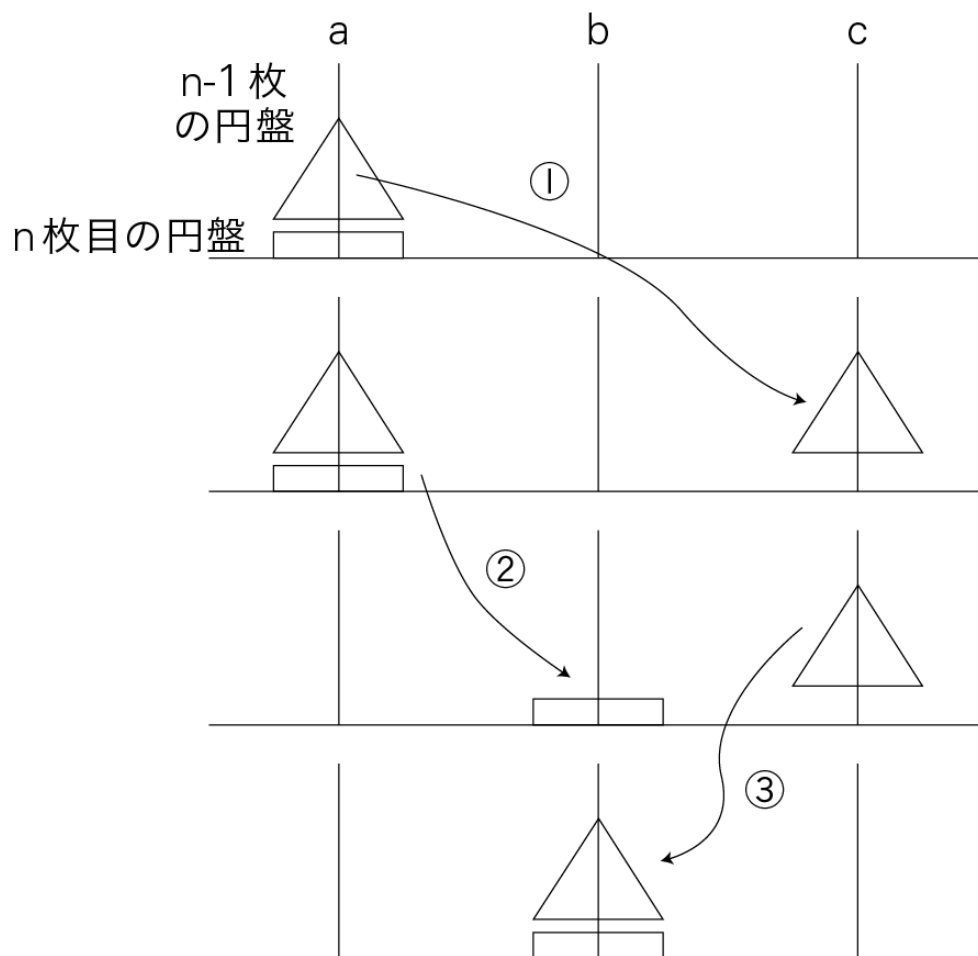
n 枚の円盤を $a \Rightarrow b$ に移す作業は、次のような作業に分解できます。①と③の作業が再帰的な作業となります。

① a の n-1 枚の円盤を $a \Rightarrow c$ に移す。

② n 枚目の円盤を $a \Rightarrow b$ に移す。

③ c の n-1 枚の円盤を $c \Rightarrow b$ に移す。

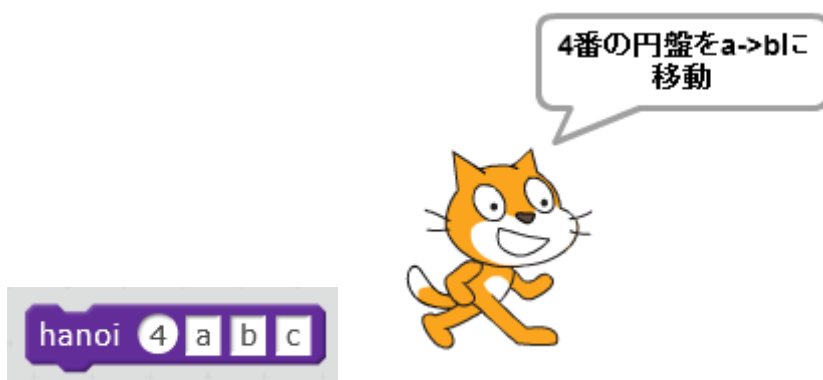
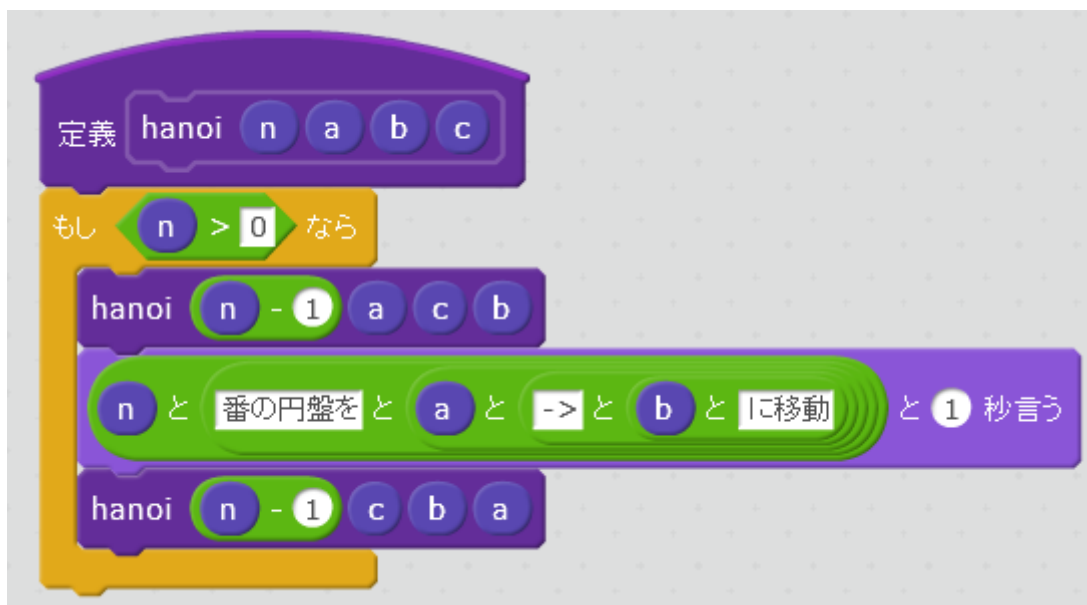
ハノイの塔



これを C プログラムで記述すると以下ようになります。

```
void hanoi(int n,char a,char b,char c)
{
    if (n>0) {
        hanoi(n-1,a,c,b);
        // n:a->b
        hanoi(n-1,c,b,a);
    }
}
```

■ハノイの塔の再帰ブロック hanoi(n,a,b,c)



6 章 Scratch の座標

Scratch の画面の座標は画面の中心を原点 (0,0) とし、右方向を x 軸の正、上方向を y 軸の正とします。座標の範囲は x が -240~240、y が -180~180 です。

この章では以下の内容について説明します。

6-1 スプライトの座標

6-2 スプライトの角度

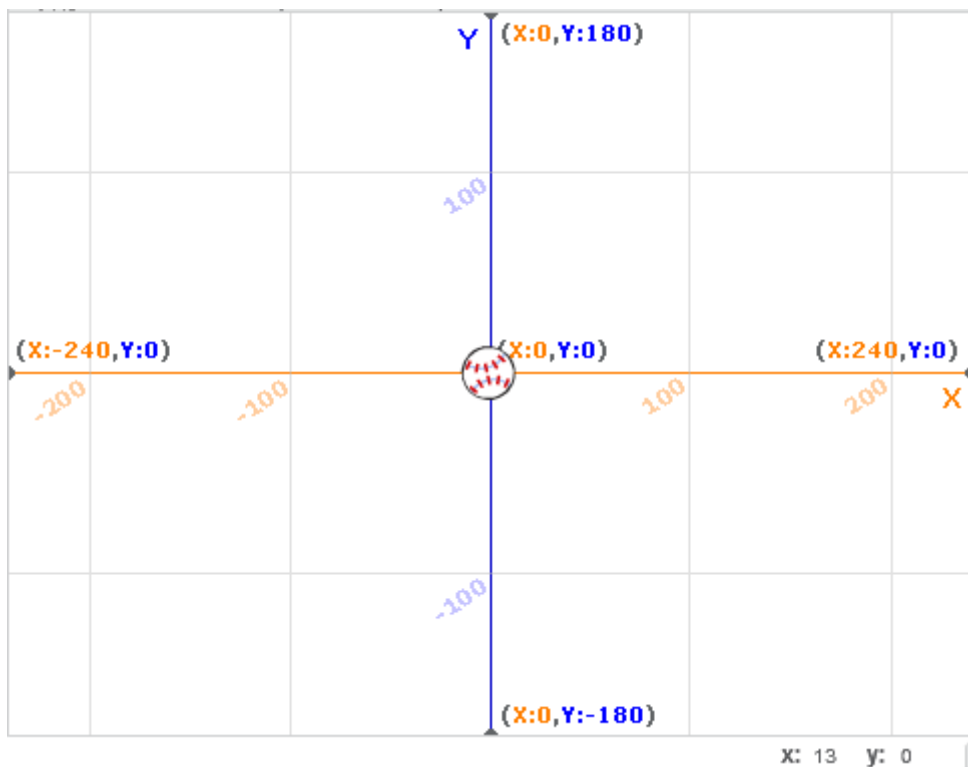
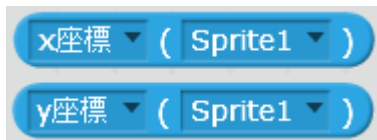
6-3 マウスの座標

6-4 関数のグラフ

6-5 ウィンドウとビューポート

6-1 スプライトの座標

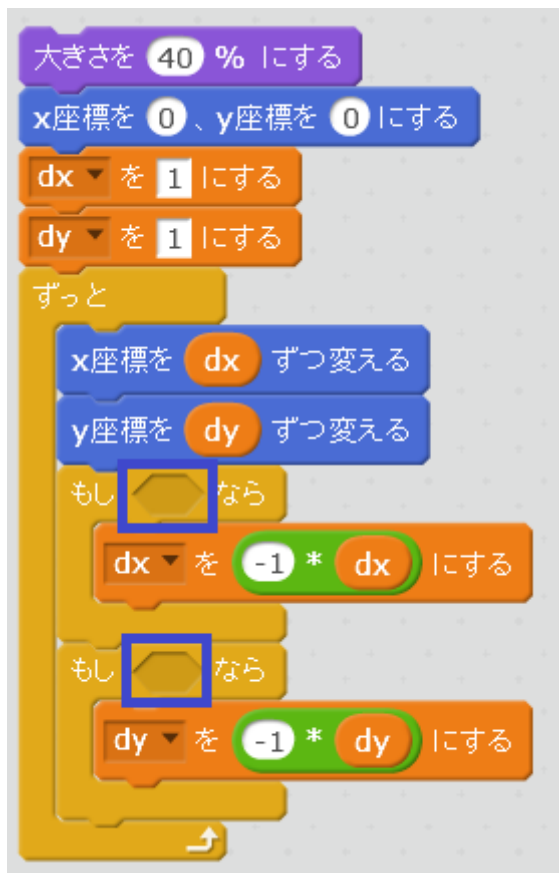
スプライトの x 座標、y 座標は以下のスクリプトで取得できます。



一般にボールを端で跳ね返すのに **もし端に着いたら、跳ね返る** を使います。

これを変数を使ってボールを移動し跳ね返すことにします。

- 変数 **dx** を x 方向の移動歩数、変数 **dy** を y 方向の移動歩数とします。初期値を 10 とします。
- スプライトの x 座標を **dx** ずつ変えます。y 座標を **dy** ずつ変えます。
- ボールの大きさを 40%にしたときのサイズを調べると、約 26 なので、ボールの中心に対して ±13 のサイズとなります。
- スプライトの x 座標が -227 未満か 227 を超えていれば、縦壁にあたったと判断し、**dx** の符号を逆にします。10 なら -10、-10 なら 10 です。
- スプライトの y 座標が -167 未満か 167 を超えていれば、横壁にあたったと判断し、**dy** の符号を逆にします。



上の青枠に以下が入ります。



7章 イベント

イベントは緑旗やスプライトをクリックした時などに発生します。イベント処理を行うスクリプトブロックとして以下があります。茶色のブロックです。



この章では以下の内容について説明します。

7-1 クリックイベント

7-2 キーイベント

7-3 背景イベント

7-4 タイマーイベント

7-1 クリックイベント

クリックイベントとして以下の2つがあります。



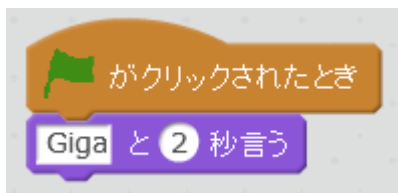
プログラム開始の緑旗がクリックされたときに発生するイベントです。全スプライトがこのイベントを同時に受信します。



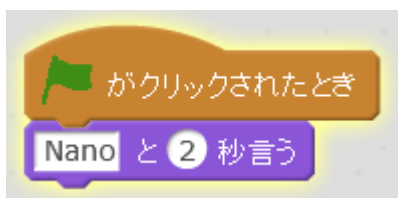
スプライトがクリックされたときに発生するイベントです。当該スプライトしか受信できません。

■緑旗イベント

・ Giga

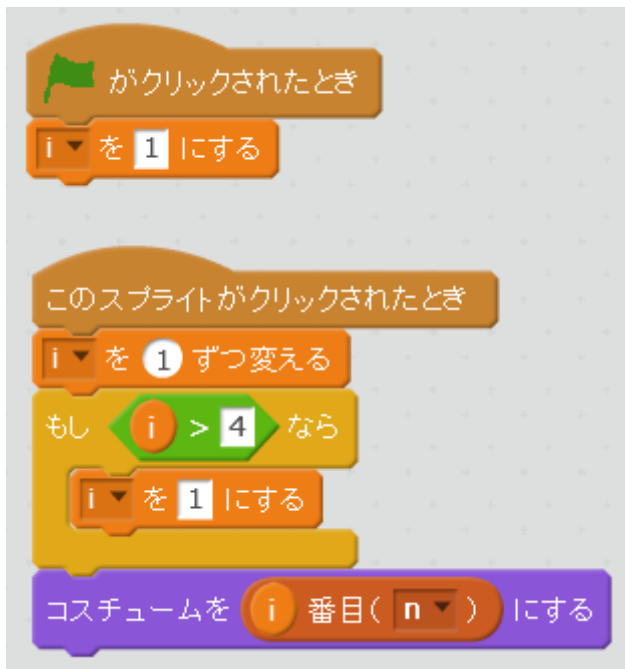


・ Nano



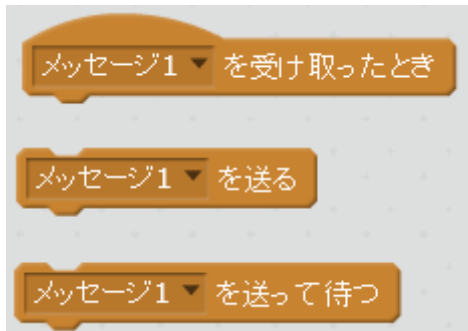
■ スプライト・クリックイベント

緑旗のクリックで変数*i*を1に初期設定します。赤矢印スプライトのコスチュームは右、左、下、上なので、コスチュームを使って矢印を反時計方向に回転するには、コスチュームを1,4,2,3番の順序で変えます。この順序データをリスト*n*に格納しておきます。赤矢印スプライトがクリックされるたびに*i*を+1して*n[i]*で示すコスチュームにします。



8 章 メッセージ

メッセージはスプライト間でメッセージを送受信することで交信を行う機能です。メッセージの送受信として以下のスクリプトがあります。イベントと同じ茶色のブロックです。



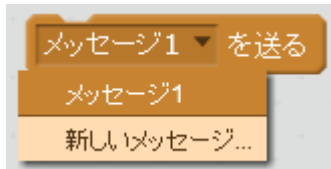
この章では以下の内容について説明します。

- 8-1 メッセージのやりとり
- 8-2 メッセージ名を変数で管理
- 8-3 開始の通知
- 8-4 判定の通知

8-1 メッセージのやりとり

■メッセージ名の登録

「新しいメッセージ」を選択し、メッセージ名を登録します。



■Giga と Gobo の会話

Giga と Gobo で以下のような会話をします。

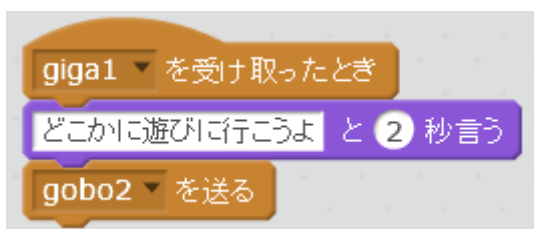
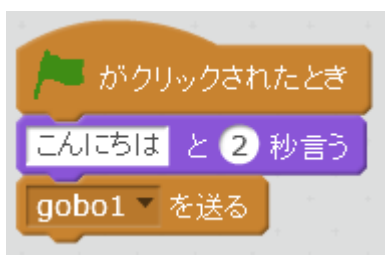
Giga : こんにちは

Gobo:やあ

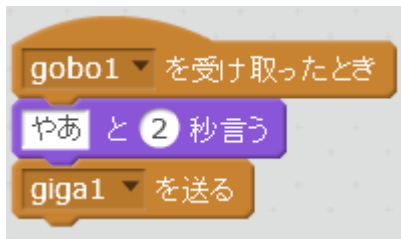
Giga : どこかに遊びに行こうよ

Gobo:海に行こう

・ Giga

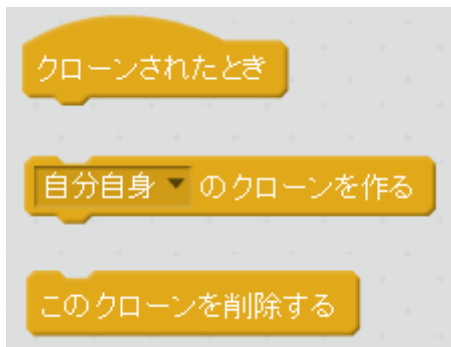


• Gobo



9 章 クローン

クローンとは複製のことです。スプライトと同じクローンを作ることができます。クローンに関するブロックは「制御」ブロックにあります。黄色のブロックです。



この章では以下の内容について説明します。

- 9-1 自分自身のクローンを作る
- 9-2 クローン生成時の通知イベント
- 9-3 クローン生成のタイムラグ
- 9-4 自分自身以外のクローンを作る
- 9-5 クリック位置のクローンを別のクローンで置き換える

9-1 自分自身のクローンを作る



指定したクローンを作ります。自分自身または他のスプライトを指定できます。

■自分自身のクローンを作る

10 歩動き、15 度回転する動作ごとに色の効果を 10 ずつ変えながら自分自身のクローンを作ります。スタンプと同じような結果になりますが、スタンプはステージへの描画で、クローンは複製されたスプライトです。



10章 調べる

マウス、スプライト、キー、日時などの情報を取得することができます。データ入力をすることもできます。これらのスクリプトは水色のブロックです。

この章では以下の内容について説明します。

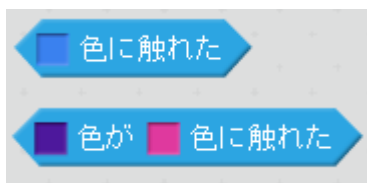
- 10-1 触れたかどうか
- 10-2 マウス情報
- 10-3 スプライトの情報
- 10-4 キーが押されたか
- 10-5 日時
- 10-6 データ入力

10-1 触れたかどうか

触れたかどうか調べるスクリプトとして以下があります。



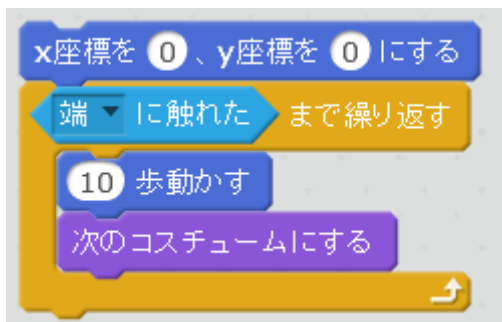
マウスポインタ、端、スプライトに触れたかどうか調べます。



色に触れたかどうか調べます。

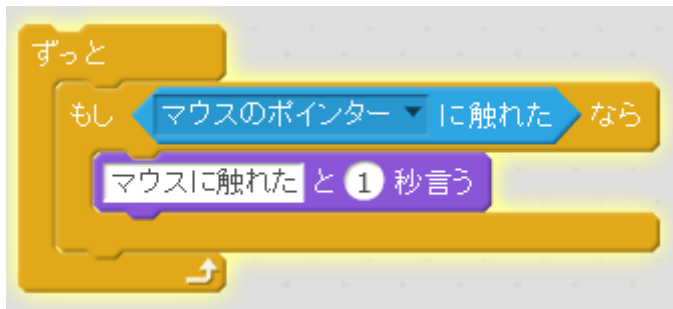
■ 端に触れたかどうか

スプライトが端に触れるまで動きます。



■マウスに触れたかどうか

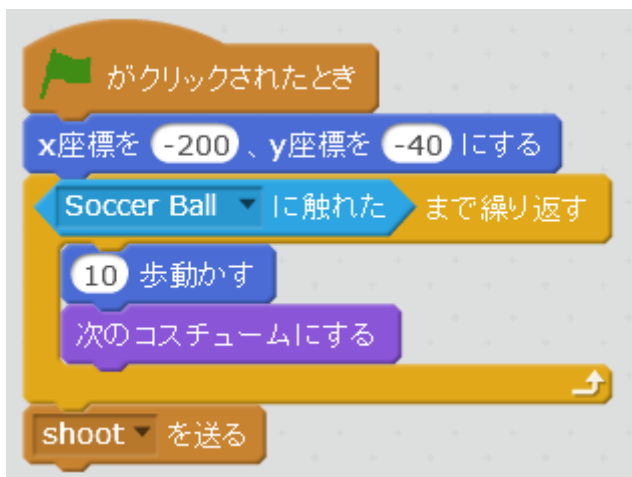
スプライトがマウスに触れたたら「マウスに触れた」と言います。



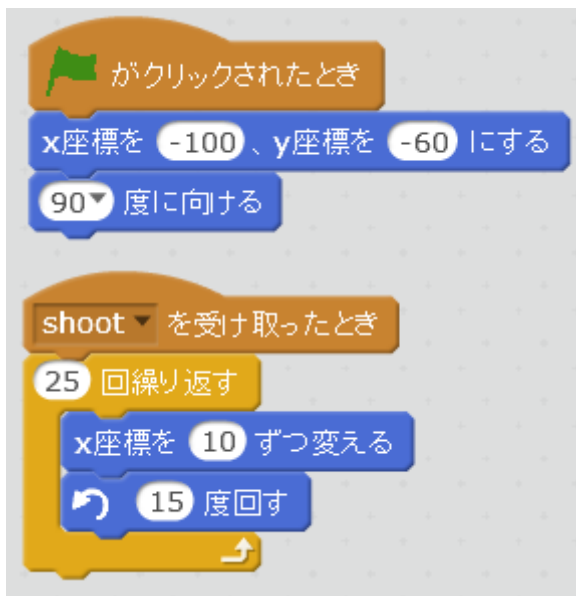
■他のスプライトに触れたかどうか

ねこ (Sprite1) がボール (Soccer Ball) に触れるまで歩き、メッセージ shoot を送ります。
ボール (Soccer Ball) はメッセージ shoot を受け取ったらゴールに向けて動きます。

・Sprite1 のスクリプト

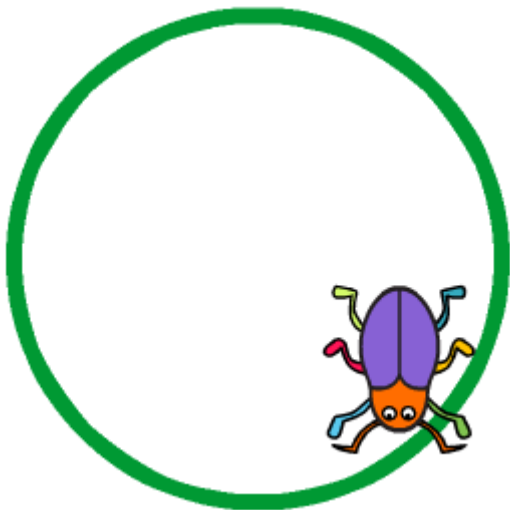


• Soccer Ball のスクリプト

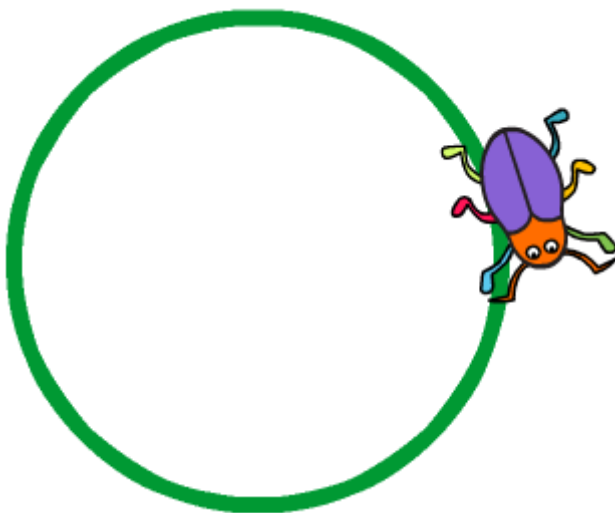
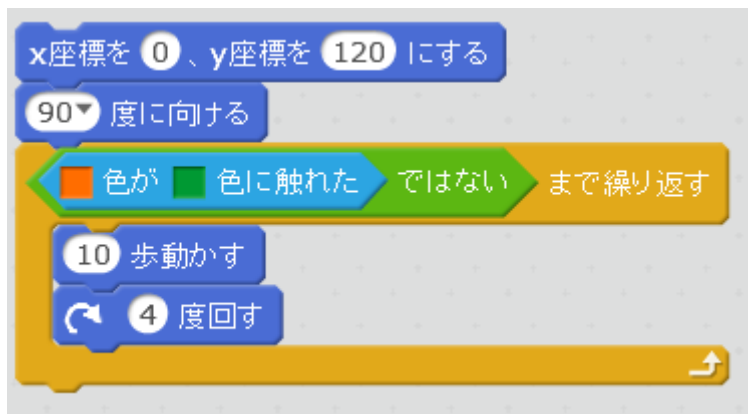


■色に触れたかどうか

スプライト（Beetle）が緑色に触れたら 30° 回します。これでスプライトは緑色の円の中を移動します。



スプライト（Beetle）の頭の赤部分が緑から離れると止まります。



1 1 章 動き

歩く、回転するなどの動きを行うスクリプトとして以下があります。青色のブロックです。
この章では以下の内容について説明します。

11-1 移動

11-2 回転

11-3 マウスへの追従

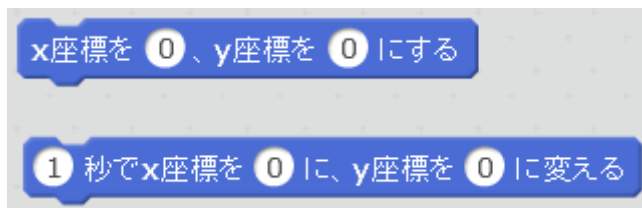
11-4 端で跳ね返る

11-1 移動

移動を行うスクリプトとして以下があります。



現在の向きに、指定した歩数動きます。



指定した x,y 座標に移動します。



指定した x 座標または y 座標に移動します。



x 座標または y 座標を指定した値だけ変えます。



スプライトの x 座標または y 座標を取得します。

■ Ghost1 と Ghost2 をマウスに追従して移動

Ghost1 は 0.2 秒遅れ、Ghost2 は 0.5 秒遅れで移動します。Ghost2 の x 座標はマウスの x 座標+20 の位置とします。



・ Ghost1



・ Ghost2



1 2 章 み た 目

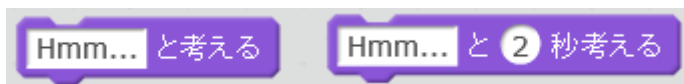
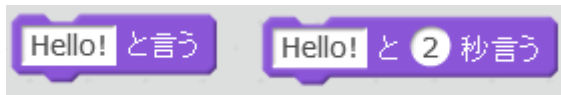
見た目の動作として、スプライトに吹き出しを出して「言う」、「考える」という動作、スプライトのコスチューム、色、大きさを変える動作、背景を変える動作などがあります。紫色のブロックです。



この章では以下の内容について説明します。

- 12-1 話す
- 12-2 表示・非表示と階層
- 12-3 コスチューム
- 12-4 画像効果
- 12-5 大きさ
- 12-6 背景
- 12-7 背景のスク립ト

12-1 話す

話すスクリプトとして以下があります。



表示する「秒」を指定している場合はその秒数がくれば表示は消えます。秒を指定しない場合は表示し続けます。それを消すには表示文字が空の  または  を使います。

13章 音

スプライトが持っている鳴き声などの音を鳴らしたり、リズム楽器を鳴らすことができます。音符を指定して楽器を鳴らすこともできます。音のスクリプトは赤紫のブロックです。この章では以下の内容について説明します。

13-1 音の登録

13-2 音を鳴らす

13-3 リズム楽器を鳴らす

13-4 楽器を音符で鳴らす

13-5 音符データをリストに格納

13-6 音量とテンポ

13-1 音の登録

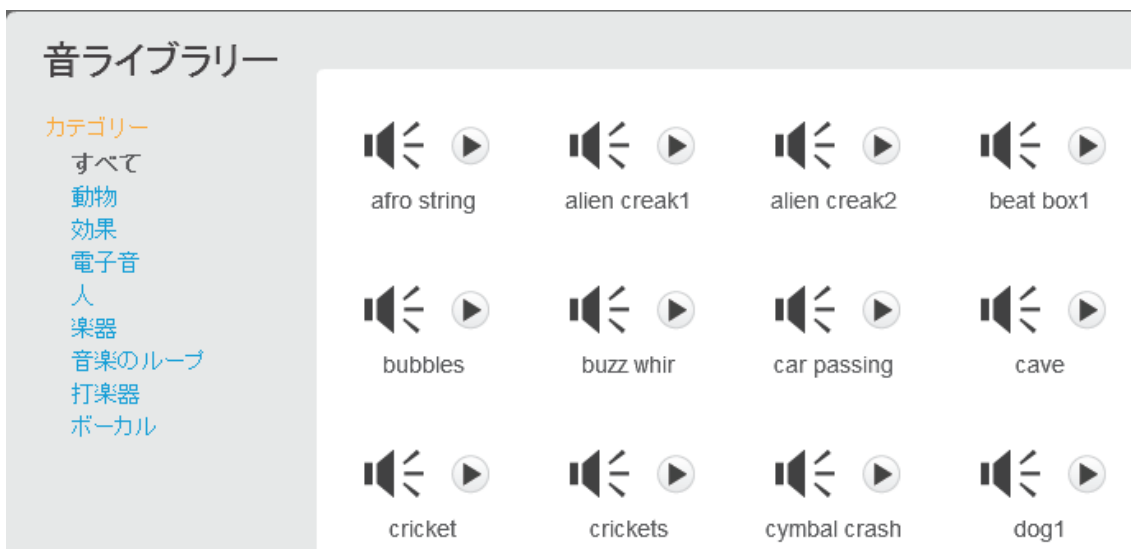
・音はスプライトごとに登録する必要があります。標準のねこには標準で **meow** という音が登録されています。



・音はライブラリ、録音、ファイルから作ることができます。



を選ぶと以下のようなライブラリが表示されますので、希望の音を登録します。





を選ぶと、録音した音を登録できます。



を選ぶと、音楽ファイルの音を登録できます。以下は sound1.mp3 ファイルから登録した音です。



1 4 章 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。ペン機能を行うスクリプトは緑色のブロックです。

この章では以下の内容について説明します。

14-1 ペン機能を行うスクリプト

14-2 スタンプ

14-3 ペンの設定ブロック

14-4 **setpoint** と **moveto**

14-5 **move**

14-6 グラフィックス用ブロック

14-7 リカーシブグラフィックス

14-1 ペン機能を行うスクリプト

ペン機能を行う主なスクリプトとして以下があります。



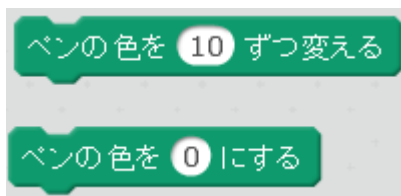
ステージに描画されているものをすべて消します。



ペンの上げ、下げを指定します。



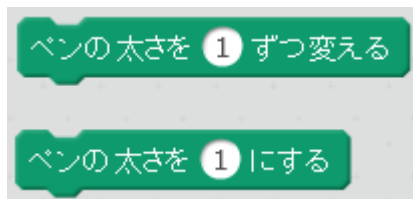
ペンの色を指定した色に設定します。色を指定するには [] をクリックするとマウスがハンドマークになるので、画面上のスプライトや、スクリプトブロックなどの色をクリックします。するとその色が選ばれます。



ペンの色を数値で指定します。

■ 色の値

色の値は 0 (赤) ~199 (赤紫) です。200 で赤に戻り、200 以上は 200 で割った余りの値が採用されます。緑は $200/3=66$ 付近、青は $200 \times 2/3=133$ 付近です。

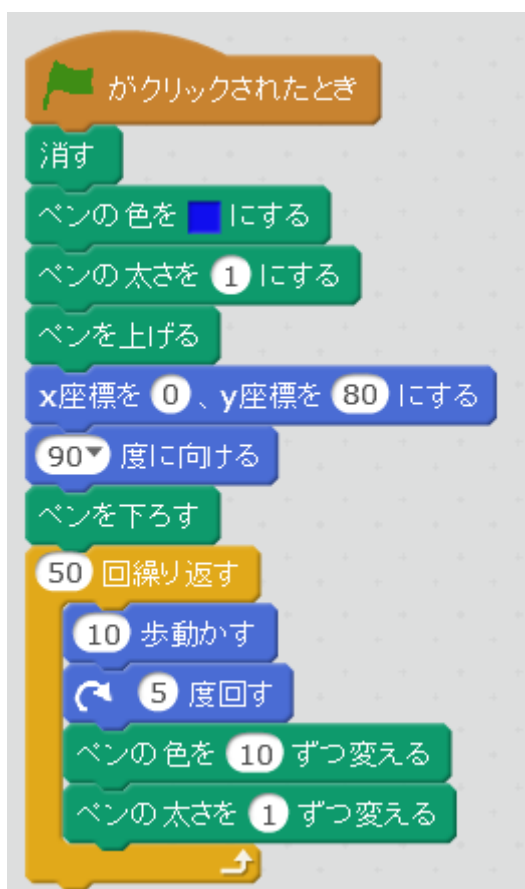


ペンの太さを指定した数値に変えます。



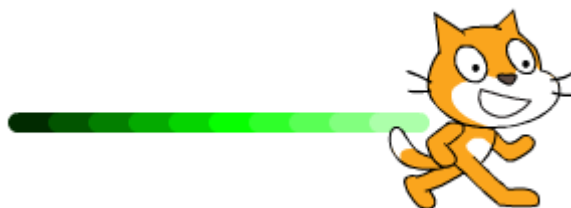
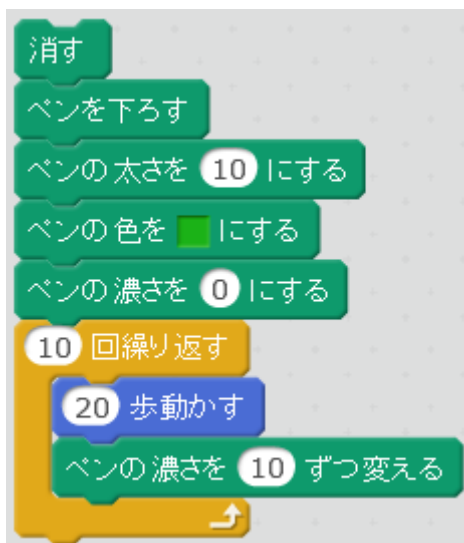
ペンの濃さを指定した数値に変えます。指定する濃さの範囲は 0～100 で中央値の 50 が標準の濃さ、0 が黒に近い濃さ、100 が白に近い濃さとなります。

■ ペンの色と太さを変える





■ ペンの濃さを変える



著者略歴

河西 朝雄（かさいあさお）

山梨大学工学部電子工学科卒（1974 年）。長野県岡谷工業高等学校情報技術科教諭、長野県松本工業高等学校電子工業科教諭を経て、現在は「カサイ．ソフトウェアラボ」代表。

「主な著書」

「入門ソフトウェアシリーズ C 言語」、「同シリーズ Java 言語」、「同シリーズ C++」、「入門新世代言語シリーズ VisualBasic4.0」、「同シリーズ Delphi2.0」、「やさしいホームページの作り方シリーズ HTML」、「同シリーズ JavaScript」、「同シリーズ HTML 機能引きテクニック編」、「同シリーズホームページのすべてが分かる事典」、「同シリーズ i モード対応 HTML と CGI」、「同シリーズ i モード対応 Java で作る i アプリ」、「同シリーズ VRML2.0」、「チュートリアル式言語入門 VisualBasic.NET」、「はじめての VisualC#. NET」、「C 言語用語辞典」ほか（以上ナツメ社）

「構造化 BASIC」、「Microsoft Language シリーズ Microsoft VISUAL C++初級プログラミング入門上、下」、「同シリーズ VisualBasic 初級プログラミング入門上、下」、「C 言語によるはじめてのアルゴリズム入門」、「Java によるはじめてのアルゴリズム入門」、「VisualBasic によるはじめてのアルゴリズム入門」、「VisualBasic6.0 入門編、中級テクニック編、上級編」、「Internet Language 改訂新版シリーズ ホームページの制作」、「同シリーズ JavaScript 入門」、「同シリーズ Java 入門」、「New Language シリーズ標準 VisualC++プログラミングブック」、「同シリーズ標準 Java プログラミングブック」、「VB.NET 基礎学習 Bible」、「原理がわかるプログラムの法則」、「プログラムの最初の壁」、「河西メソッド：C 言語プログラム学習の方程式」、「基礎から学べる VisualBasic2005 標準コースウェア」、「基礎から学べる JavaScript 標準コースウェア」、「基礎から学べる C 言語標準コースウェア」、「基礎から学べる PHP 標準コースウェア」、「なぞりがき C 言語学習ドリル」、「C 言語標準ライブラリ関数ポケットリファレンス[ANSI C,ISO C99 対応]」、「C 言語 標準文法ポケットリファレンス[ANSI C,ISOC99 対応]」、「[標準] C 言語重要用語解説 ANSI C / ISO C99 対応」ほか（以上技術評論社）

「電子書籍：カサイ．ソフトウェアラボ」

「Android プログラミング Bible 初級 基礎編」、「Android プログラミング Bible 中級 Android 的プログラミング法」、「Android プログラミング Bible 上級 各種処理」、「Android プログラミング完全入門」、「iPhone&iPad プログラミング Bible[上]」、「iPhone&iPad プログラミング Bible[下]」、「JavaScript によるはじめてのアルゴリズム入門」、「Web アプリ入門 (HTML5+JavaScript)」、「HTML5 を使った JavaScript 完全入門」、「Scratch プログラミング入門」、「小・中学生のための Scratch プログラミング入門」、「ideon で学ぶ小・中学生のためのプログラミング入門 C 言語編」、「同 Java 言語編」



Scratch 完全攻略マニュアル

Scratch プログラミング技法を学ぶ

2018 年 1 月 20 日 初版 第 1 刷

著者＝河西 朝雄

発行者＝河西 朝雄

発行所＝カサイ．ソフトウェアラボ

長野県茅野市ちの 813 TEL.0266-72-4778

表紙デザイン＝河西 朝樹

本書の一部または全部を著作権法の定める範囲を超え、無断で複写、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。

定価＝1,500 円＋税

©2018 河西 朝雄