
Scratch
アルゴリズム辞典
河西　朝雄著
定価　1,500円＋税

Scratch アルゴリズム辞典

河西　朝雄著

はじめに

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）という。プログラム処理では多量のデータを扱うことが多く、この場合、取り扱うデータをどのようなデータ構造（data structure）にするかで、問題解決のアルゴリズムが異なってくる。データ構造とアルゴリズムは密接な関係にあり、良いデータ構造を選ぶことが良いプログラムを作ることにつながる。プログラムの世界に特有なアルゴリズムとして再帰という考え方がある。再帰とデータ構造の木を組み合わせることで効率的なアルゴリズムができる。

ある問題を解くためのアルゴリズムは複数存在するが、人間向きのアルゴリズムが必ずしもコンピュータ向きのアルゴリズムにならないこともある。本書では先達が研究した伝統的なアルゴリズムやScratch特有のアルゴリズムをできるだけ多く集め、以下の１２のカテゴリで解説する。

また、アルゴリズムの難易度に応じて初級　中級　上級のマークを付けた。

1章　図形アルゴリズム
2章　アニメーションアルゴリズム
3章　画面出力アルゴリズム
4章　Scratch特有のアルゴリズム
5章　リスト操作アルゴリズム
6章　データ処理アルゴリズム
7章　文字列処理アルゴリズム
8章　数値処理アルゴリズム
9章　再帰アルゴリズム
10章　データ構造アルゴリズム
11章　ゲームアルゴリズム
12章　ミスレニアス・アルゴリズム

2018年8月1日

河西　朝雄

目次

1章　図形アルゴリズム

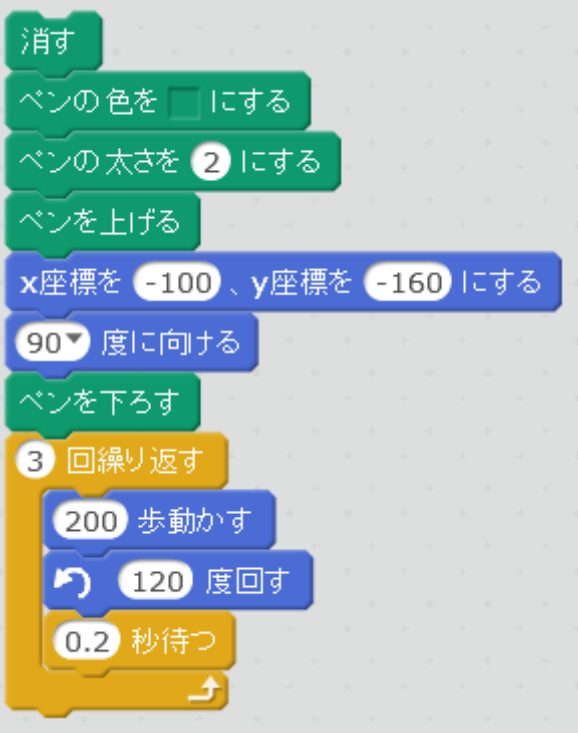
スプライトの移動、回転の軌跡をペン機能を使って描画することができる。図形の描画に関し以下の内容を説明する。

リカーシブグラフィックについては10章「10-7　リカーシブグラフィックス」で説明する。

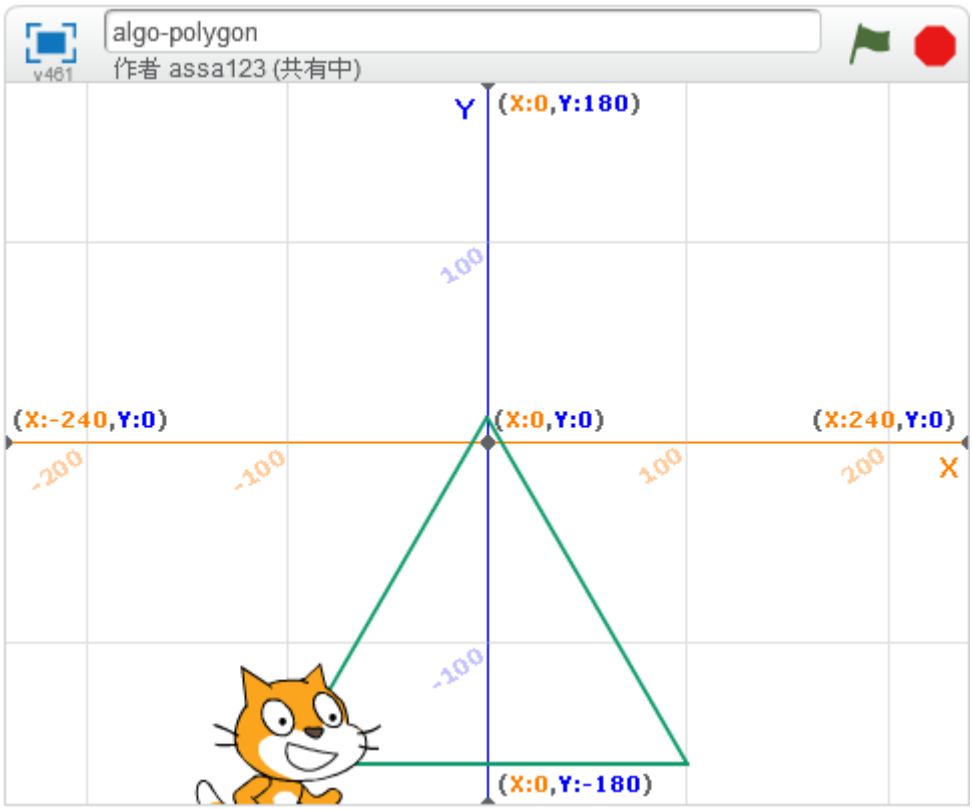
1-1　多角形を描く
1-2　渦巻き模様
1-3　円周上の点を結んでできる図形
1-4　グラフを描く
1-5　タートルグラフィックス
1-6　図形描画メソッド
1-7　ウィンドウとビューポート
1-8　幾何学模様
1-9　２次元座標変換
1-10　３次元座標変換
1-11　ワイヤーフレームモデル
1-12　回転体モデル
1-13　３次元関数
1-14　陰線処理

1-1　多角形を描く 初級

■三角形



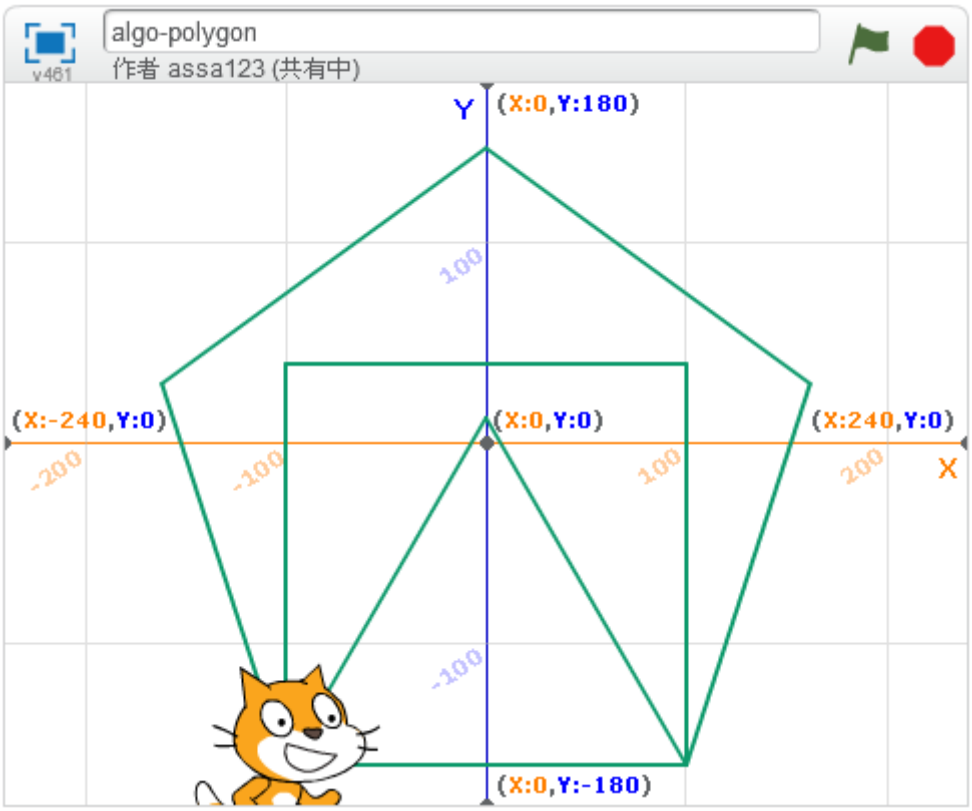
・始点を（-100,-160）に移し、右方向を向く。
・200 歩移動し、120° 反時計回りに回転。
・これを 3 回くり返す。



■四角形、五角形

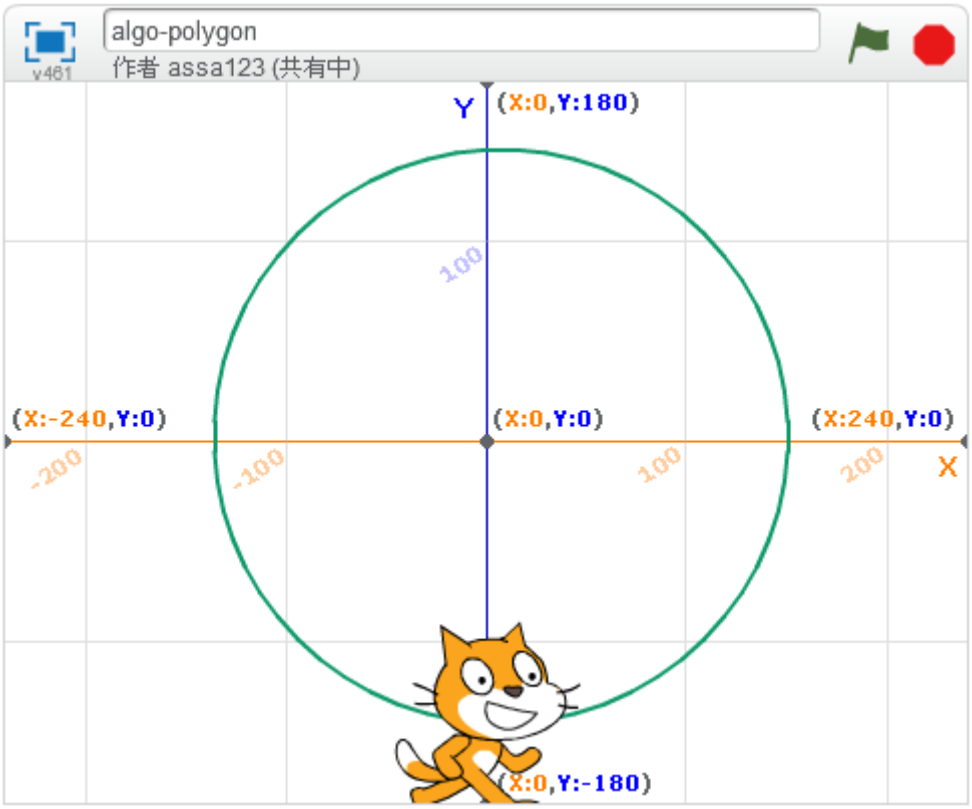
・四角形は 90° 回転で 4 回くり返す。

・五角形 72° 回転で 5 回くり返す。



■60 角形

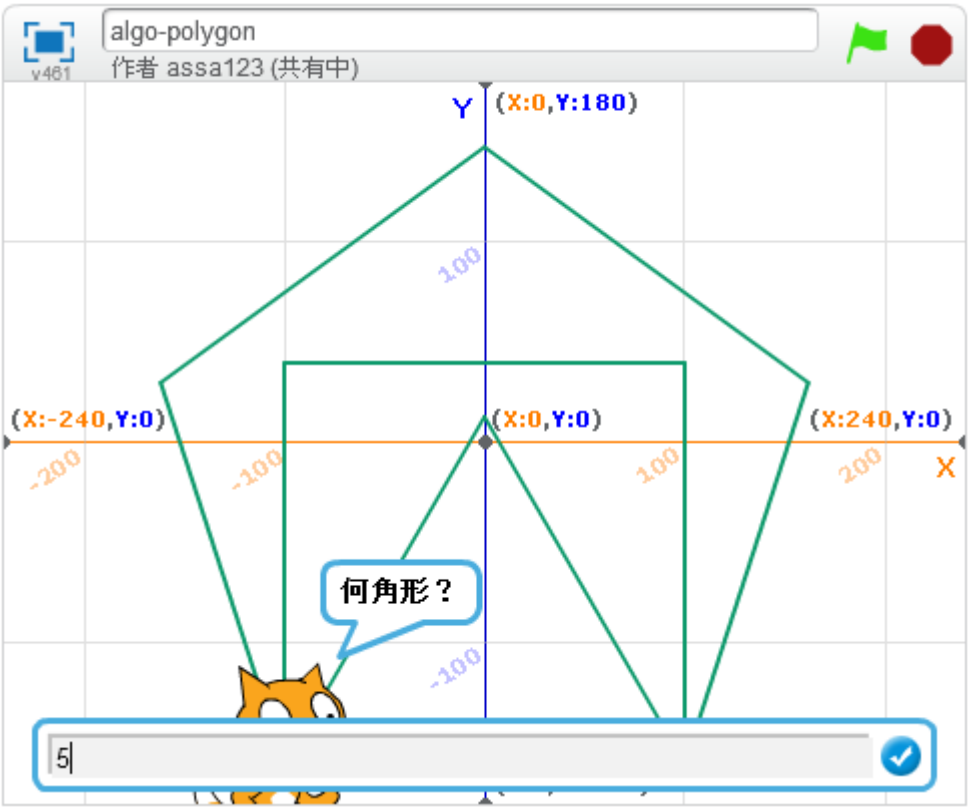
角数が上がると円に近付く。


algo-polygon
v461
作者 assa123 (共有中)
Y
(X:0,Y:180)
100
(X:-240,Y:0)
(X:0,Y:0)
(X:240,Y:0)
-200
-100
100
200
X
(X:0,Y:-180)

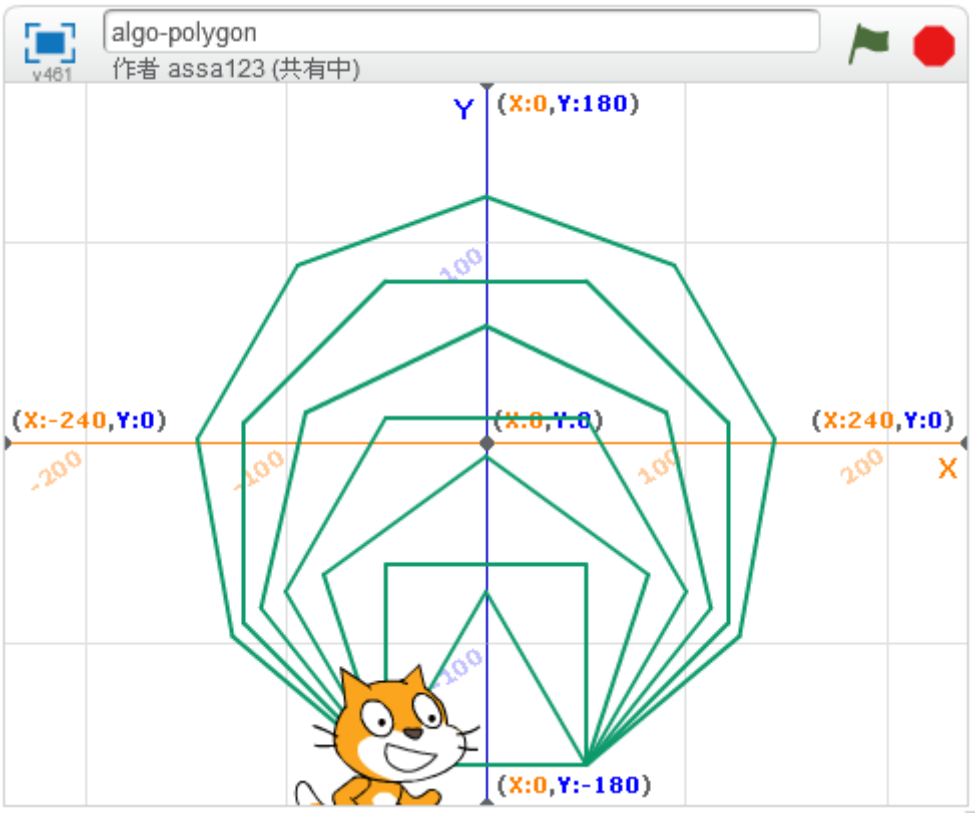
■ n 角形

・n 角形を描くときの回転角は「360/n」で求められる。

・何角形かを聞いて、その角数の多角形を描く。


algo-polygon
作者 assa123 (共有中)
Y
(X:0,Y:180)
100
(X:-240,Y:0)
(X:0,Y:0)
(X:240,Y:0)
-200
-100
100
200
X
何角形？
-100
5

・三角形~九角形までを描く。



■星型多角形

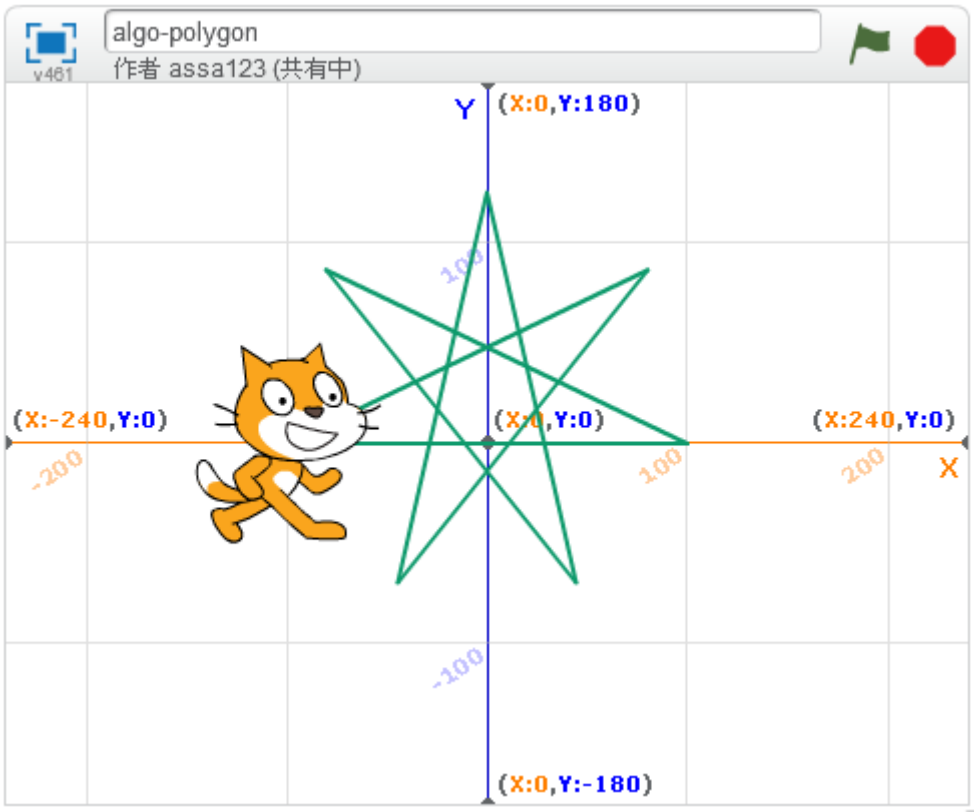
・一筆書きで描ける星型多角形は 5 角形、 7 角形、 9 角形・・・の奇数多角形である。

・星型 n 角形の回転する角度は次の式で求めることができる。

180-180/n

たとえば n が 5 なら「180-180/5=180-36=144」となる。


消す
ペンの色を にする
ペンの太さを 2 にする
ペンを上げる
x座標を -100 、y座標を 0 にする
90 度に向ける
ペンを下ろす
n を 7 にする
n 回繰り返す
200 歩動かす
180 - 180 / n 度回す
0.2 秒待つ


algo-polygon
作者 assa123 (共有中)
Y
(X:0,Y:180)
100
(X:-240,Y:0)
(X:0,Y:0)
(X:240,Y:0)
-200
100
200
X
-100
(X:0,Y:-180)

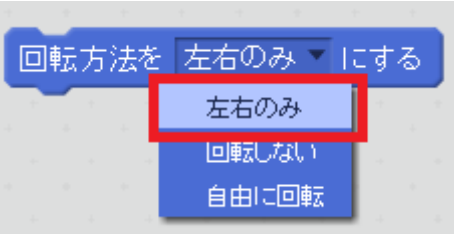
2章　アニメーションアルゴリズム

視覚的に動きのあるアニメーションに関し以下の内容を説明する。

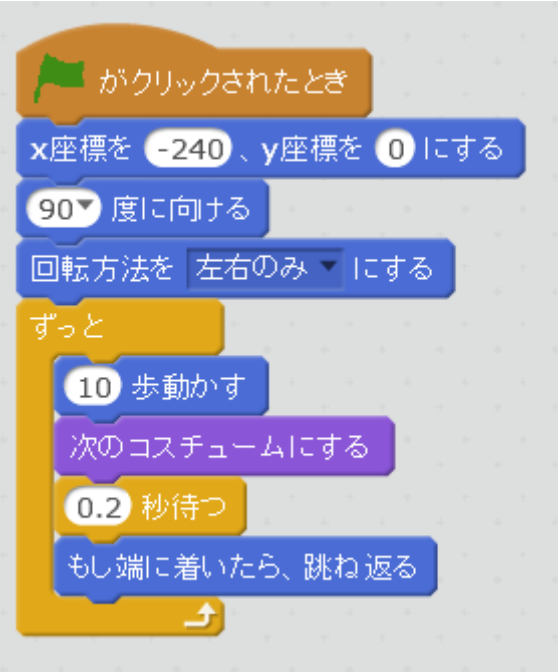
2-1　こうもりが舞う
2-2　虫たちのの運動会
2-3　ボール移動
2-4 ライントレース
2-5　7セグメント
2-6　ドットアート
2-7　数字の整列
2-8　マス状に配置
2-9　リング状に配置
2-10　回転
2-11　マウスに追従して動く
2-12　マウスに触れた/触れない
2-13　アナログ時計
2-14　デジタル時計
2-15　スクロール
2-16　砲台から玉を打つ

2-1　こうもりが舞う 初級

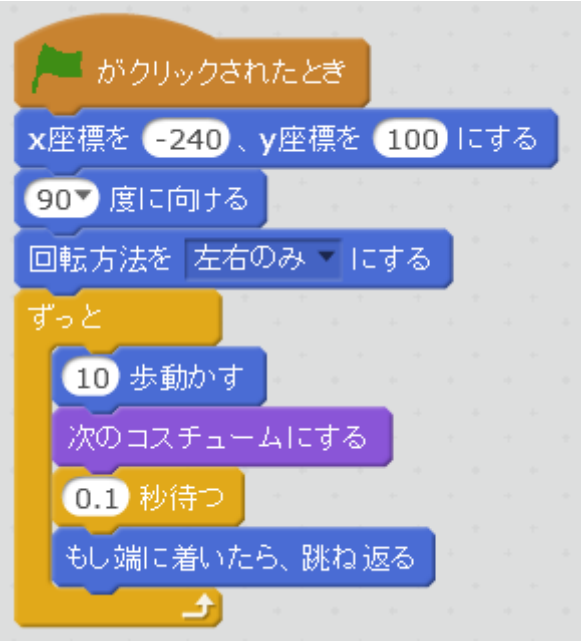
・こうもりを画面の左から右へコスチュームを変えながら移動する。
・跳ね返った時にこうもりの上下が逆にならないようにするため、「回転方法を左右のみにする」を「ずっと」の前に置く。
・待ち時間を「0.1 秒」と「0.2 秒」して移動スピードを変える。
・速さを変えるには「歩数」を変える方法と「待ち時間」を変える方法とがある。

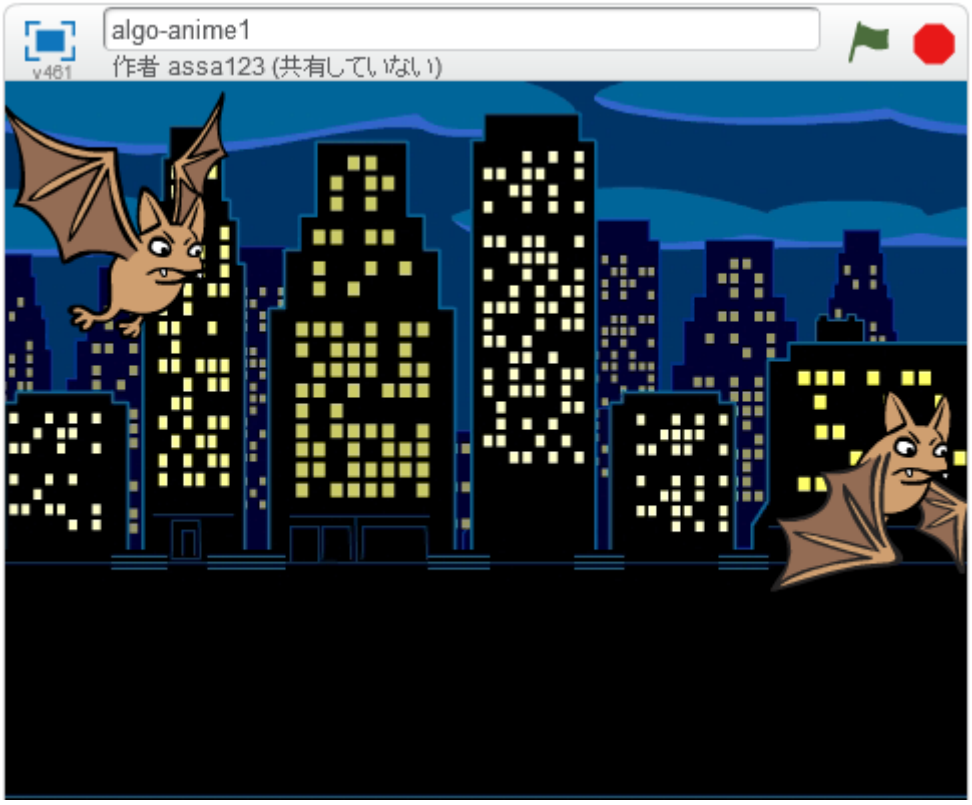


・Bat1



・Bat2


がクリックされたとき
x座標を -240 、y座標を 100 にする
90 度に向ける
回転方法を 左右のみ にする
ずっと
10 歩動かす
次のコスチュームにする
0.1 秒待つ
もし端に着いたら、跳ね返る


algo-anime1
v461
作者 assa123 (共有していない)

3 章　画面出力アルゴリズム

通常のプログラミング環境と異なり Scratch には print のような画面出力を行う命令がない。そこで output という名前のリストを出力画面の代用にし、そこに結果を表示するための init、println というユーザーブロックを作る。

また、0~9 の数字イメージを使って、指定位置に数字を表示する number というユーザーブロックを作る。。

3-1　文字の出力

3-2　書式付き出力

3-3　数字イメージの表示（大きいサイズ）

3-4　数字イメージの表示（小さいサイズ）

☆汎用ユーザブロックの変数名

いろいろな場所で利用される汎用ユーザブロックはメインプログラムの変数との重複使用を避けるための方策をとっておくことがトラブルの発生を防げる。

変数 i はしばしばくり返しで利用される。これをブロックでも使うと、以下のような場合、呼び出し側（メイン）側の i とブロック側の i が同じなので、思わぬ結果を引き起こす。

メイン側で i を使わず j などを使わんければよいのだが、いちいちブロック側の変数を考慮するのは煩雑である。そこで、この章で説明するような、汎用ユーザブロックで使用する変数は x_x のように間に「_」を入れた変数名を使用することにする。

3-1　文字の出力 初級

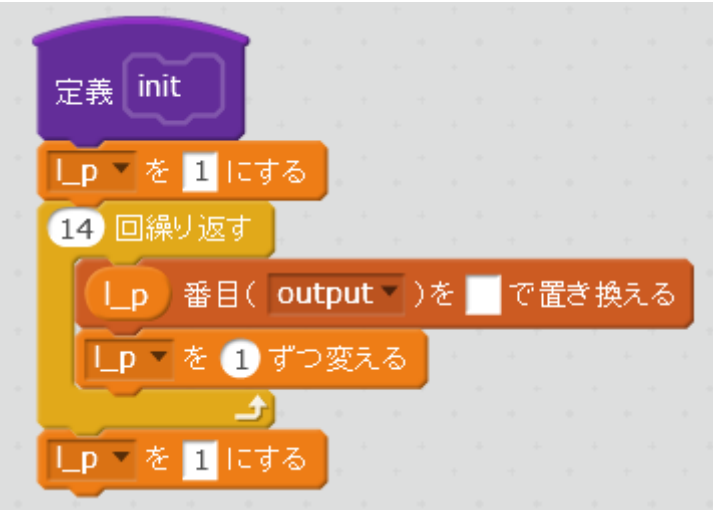
　通常のプログラミング環境と異なり Scratch には print のような画面出力を行う命令がない。そこで output という名前のリストを出力画面の代用にし、そこに結果を表示するための init、println というユーザーブロックを作る。

・output という名前のリストを作成し、表示状態にする。要素数を 14 とし画面一杯に幅を広げて表示する。

・output への表示位置は変数 l_p で管理する。

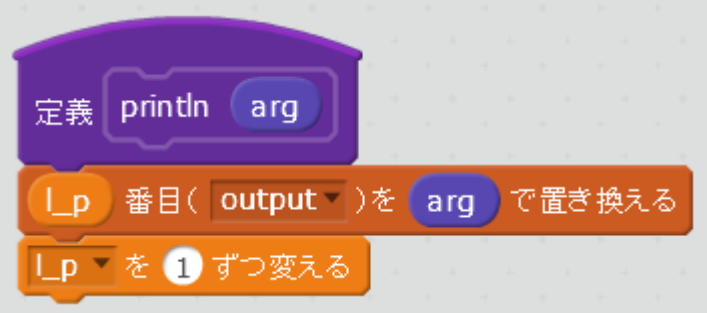
■init

init はリスト output の内容をクリアする。



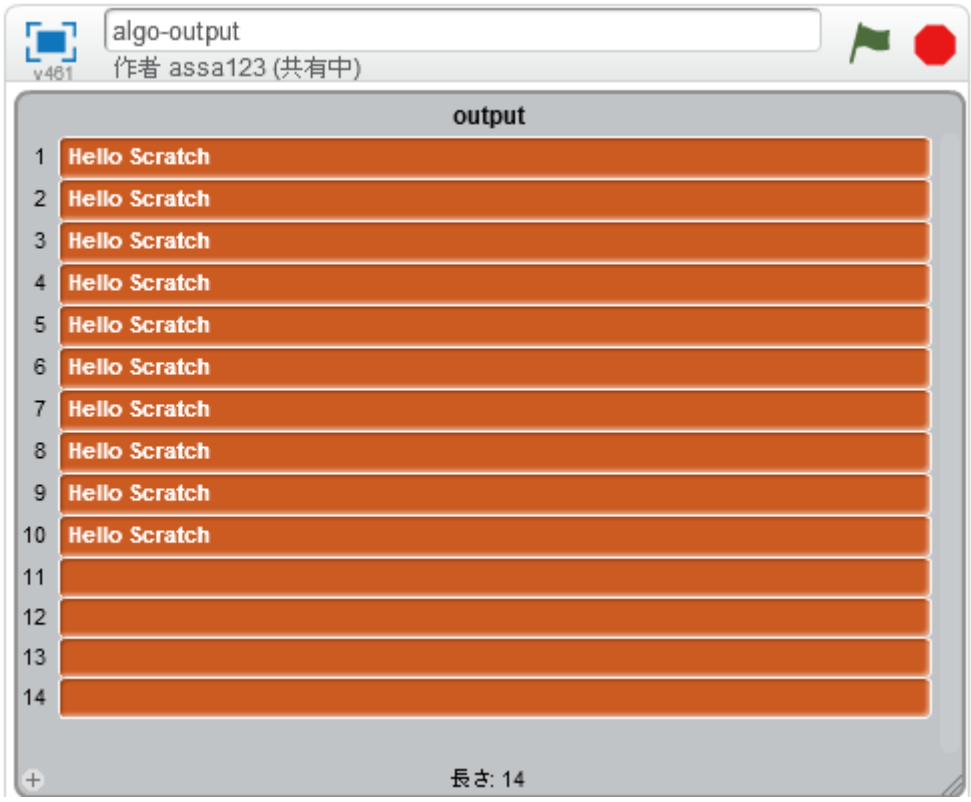
■println(arg)

println(arg)は引数 arg の内容を画面に表示し、改行（次のリスト要素に進める）する。



■println の例

「Hello Scratch」を 10 回表示する。



4 章　Scratch 特有のアルゴリズム

Scratch は他の一般的な言語とは異なる部分がある。制御構造に制約はあるが、クローンやサウンドなど Scratch ならではの機能がある。この章では以下の Scratch 特有のアルゴリズムについて説明する。

4-1　制御構造 初級

Scratch には一般の言語が持つ do while 文、else if 文、switch case 文、break 文がないので、その代用法を説明する。

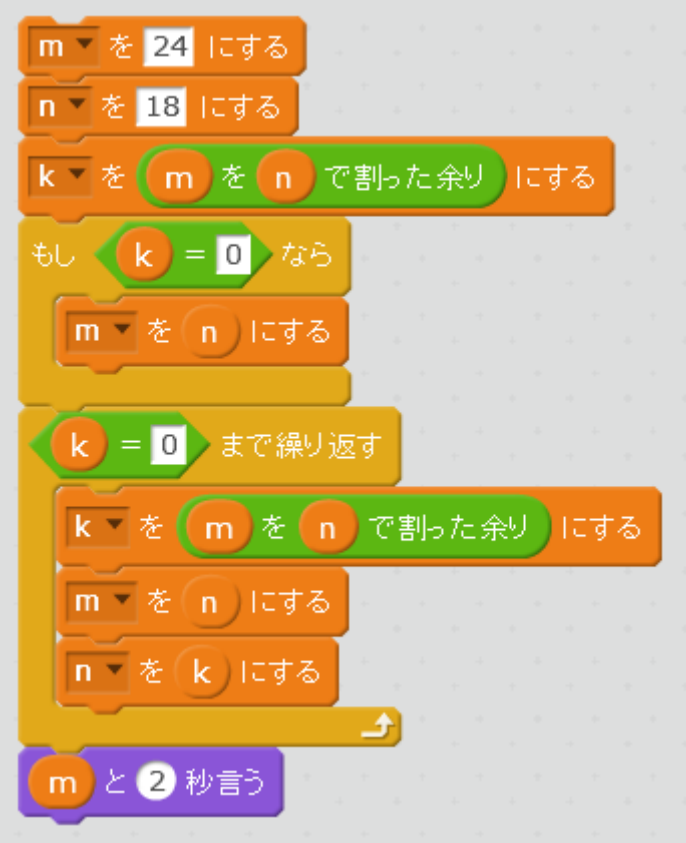
■do while 文

Scratch には do while 文に相当するものがない。do while 文はくり返し条件の前に文を 1 回実行するで、 

の前に最初の 1 回の実行文と判定を入れる。

```
m=24;n=18;
do {
      k=m % n;
      m=n;
      n=k;
} while (k!=0);
```

は以下のようになる。条件の「k!=0」は逆の「k=0」になる。



■else if 文

Scratch には多方向分岐を行う else if 文はないので、 

の「でなければ」に

をネストする。

```
a=10;
if (a<0) {
    // 負
}
else if (a==0) {
    // ゼロ
}
else {
    // 正
}
```

は以下のようになる。

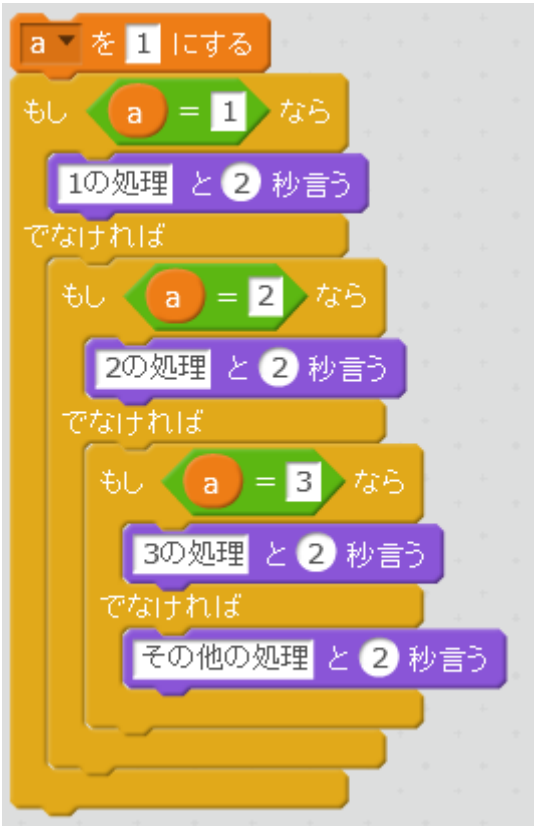
■switch case 文

Scratch には多方向分岐を行う switch case 文はないので、else if 文と同様に

をネストする。

```
a=1;
switch (a) {
      case 1: // 1 の処理;break;
      case 2: //2 の処理;break;
      case 3: //3 の処理;break;
      default://その他の処理
}
```

は以下のようになる。

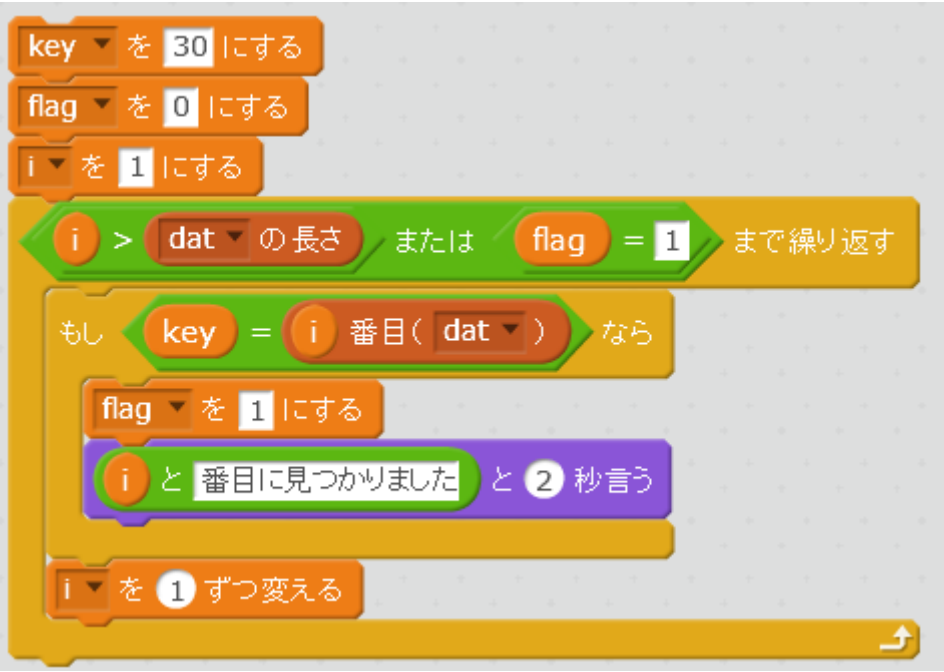


■break 文

Scratch には break 文はない。そで変数 flag を使って、ループから脱出する条件を作る。

```
int dat[]={10,20,30,40,50};
key=30;
for (i=0;i<a.length;i++){
      if (dat[i]=key) {
            // 見つかりました
            break;
      }
}
```

は以下のようになる。break を実行する代わりに「flag=1」という終了条件を設定する。



5章　リスト操作アルゴリズム

変数には１つのデータしか入らないが、リストには多くのデータを格納することができる。一般のプログラミング言語では配列と呼んでいる。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する。

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

5-1　リスト要素の参照 初級

リストａのｉ番目の要素は 

で取得できる。

・リストａの先頭の要素から最後の要素まで順次表示する。

6章　データ処理アルゴリズム

リストに格納された多量のデータを効率的にソートしたり探索するデータ処理アルゴリズムを紹介する。

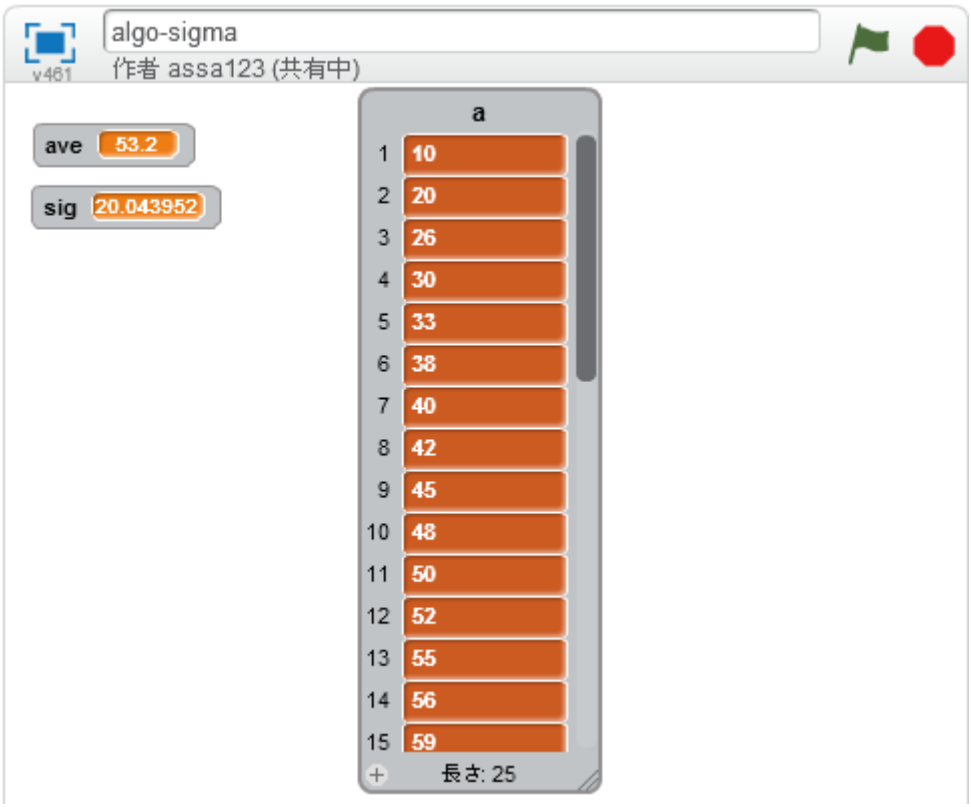
6-1　平均と標準偏差 初級

標準偏差 σ は次式で得られる。$\bar{x}$ は平均、n はデータ数、x_i は各データである。

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

リスト a に格納されているデータの平均を ave、標準偏差を sig に求める。


がクリックされたとき
n を a の長さ にする
sum を 0 にする
i を 1 にする
n 回繰り返す
sum を i 番目(a) ずつ変える
i を 1 ずつ変える
ave を sum / n にする
sum を 0 にする
i を 1 にする
n 回繰り返す
sum を i 番目(a) - ave * i 番目(a) - ave ずつ変える
i を 1 ずつ変える
sig を 平方根 (sum / n) にする


algo-sigma
v461
作者 assa123 (共有中)
ave 53.2
sig 20.043952
a
1 10
2 20
3 26
4 30
5 33
6 38
7 40
8 42
9 45
10 48
11 50
12 52
13 55
14 56
15 59
長さ: 25

7章　文字列処理アルゴリズム

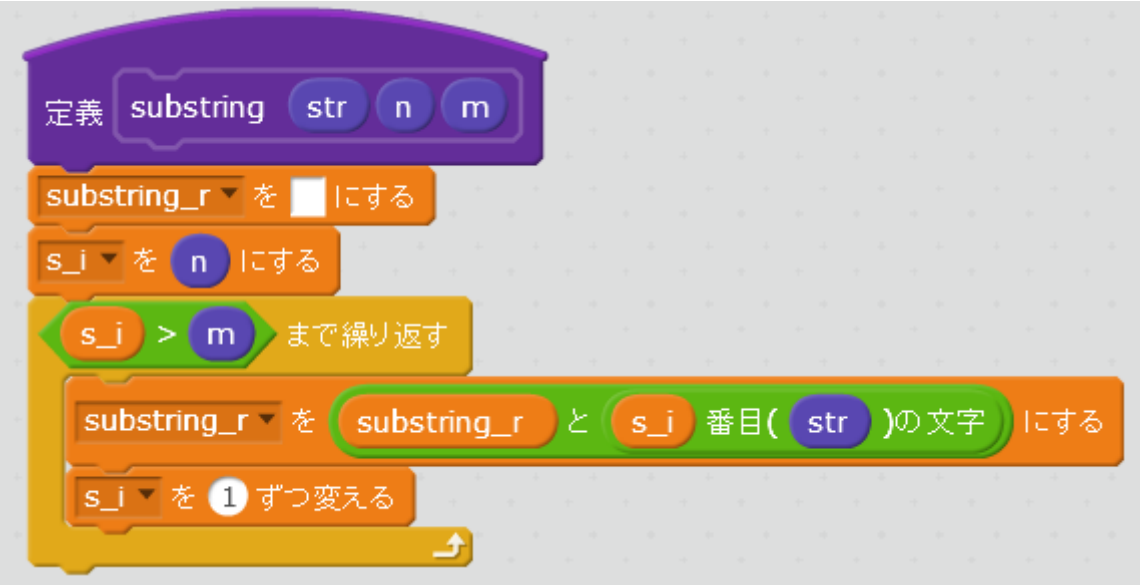
コンピュータで扱うデータをおおざっぱに分けると、数値データと文字列データである。この章では文字列データを処理する各種アルゴリズムを紹介する。

7-1　部分文字列の取得 初級

部分文字列を取得するユーザーブロックを作る。

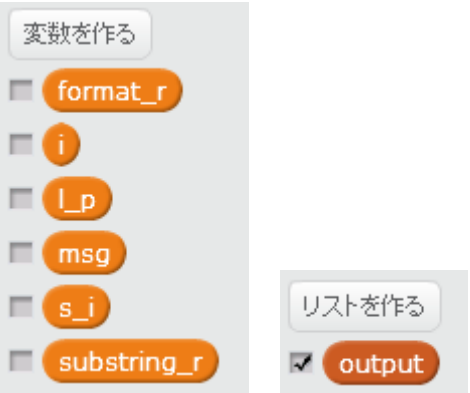
■substring(str,n,m)

str で与えられた文字列の n 番目の文字からm番目の文字までを切り取る。切り取った部分文字列は substring_r に格納されている。



■切り取った文字列の表示

「Hello Scratch」という文字列を後ろから１文字、２文字、３文字・・・と切り取り表示する。




がクリックされたとき
init
msg を Hello Scratch にする
i を 1 にする
i > msg の長さ まで繰り返す
substring msg msg の長さ - i + 1 msg の長さ
println substring_r 0
i を 1 ずつ変える


algo-substring
v461
作者 assa123 (共有中)
output
1 h
2 ch
3 tch
4 atch
5 ratch
6 cratch
7 Scratch
8 Scratch
9 o Scratch
10 lo Scratch
11 llo Scratch
12 ello Scratch
13 Hello Scratch
14
長さ: 14

8章　数値処理アルゴリズム

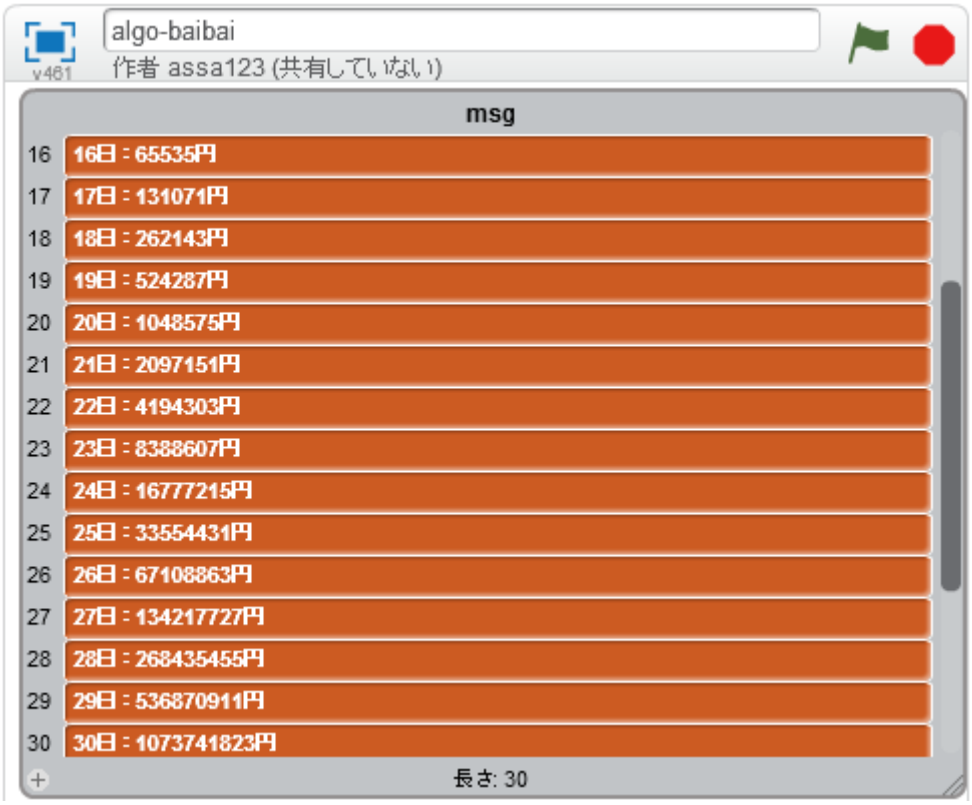
コンピュータで扱うデータをおおざっぱに分けると、数値データと文字列データである。この章では数値データを処理する各種アルゴリズムを紹介する。

8-1　倍々ゲーム 初級

１日目に１円預金する、２日目は倍の２円、３日目はさらに倍の４円、４日目はさらに倍の８円と預金していくとき、何日目で預金総額は 100 万円を越えるか調べるプログラムを作る。日を day、その日の預金額を money、預金総額を sum で管理する。

同様に、親から一ヶ月のお小遣いをもらうのに１日目に１円もらい、２日目は倍の２円、３日目はさらに倍の４円、４日目はさらに倍の８円ともらっていくとき、１ヶ月（30 日）ではいくらになるかを求めるプログラムを作る。結果は、30 日で「1,073,741,823」となる。10 億７千万。


がクリックされたとき
msg の長さ 回繰り返す
1 番目を msg から削除する
day を 1 にする
sum を 0 にする
money を 1 にする
30 回繰り返す
sum を money ずつ変える
day と 日: と sum と 円 を msg に追加する
money を money * 2 にする
day を 1 ずつ変える


algo-baibai
v461
作者 assa123 (共有していない)
msg
16 16日 : 65535円
17 17日 : 131071円
18 18日 : 262143円
19 19日 : 524287円
20 20日 : 1048575円
21 21日 : 2097151円
22 22日 : 4194303円
23 23日 : 8388607円
24 24日 : 16777215円
25 25日 : 33554431円
26 26日 : 67108863円
27 27日 : 134217727円
28 28日 : 268435455円
29 29日 : 536870911円
30 30日 : 1073741823円
長さ: 30

9章　再帰アルゴリズム

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あるがプログラムの世界では重要な考え方である。

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くり返す構造である。このような構造を一般に再帰と呼んでいる。

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）と呼ぶ。

再帰では一般に再帰からの脱出口を置かなければいけない。これがないと、再帰呼び出しが永遠に続いてしまうことになる。

再帰を用いると、複雑なアルゴリズムを明解に記述することができる。ここではハノイの塔や迷路を説明する。

9-1　ユークリッドの互除法の再帰版 初級

ユークリッドの互除法においては m-n を用いるより m mod n を用いた方が効率が良い。mとnの最大公約数を求めるユーザブロックを gcd(m,n)とすると、

m≠n なら　　gcd(m,n)=gcd(n,m mod n)
n=0 なら　　gcd(m,n)=m

ということになる。

■再帰と戻り値

以下は一般のプログラミング言語で記述した最小公倍数を求める再帰関数である。

```
int gcd(int m,int n)
{
    if (n==0)
        return m;
    else
        return gcd(n,m % n);
}
```

一般のプログラミング言語では関数なり手続きは戻り値というものを持っているが、Scratch のブロックには戻り値という概念はない。そこでブロックの戻り値は変数を使って仮想的に値を返すようにする。

■ユークリッドの互除法の再帰版

最大公約数を求める再帰ブロック gcd(m,n)を作る。結果（戻り値）は gcd_r に格納される。


がクリックされたとき
m を 128 にする
n を 72 にする
gcd m n
m と と と n と 最大公約数は と gcd_r と言う
定義 gcd m n
もし n = 0 なら
gcd_r を m にする
でなければ
gcd n m を n で割った余り


algo-reuclod
v461
作者 assa123 (共有中)
128と72最大公約数は8

10 章　データ構造アルゴリズム

コンピュータ言語が持つデータ型には数値型、文字列型などの基本データ型とデータをまとめて管理するリストがある。しかし、このようなコンピュータ言語が持つデータ型だけでは、大量のデータや複雑な構造のデータを効率よく操作することはできない。そこでデータ群を都合よく組織化するための抽象的なデータ型をデータ構造と呼ぶ。代表的データ構造として以下がある。

・表（table：テーブル）
・棚（stack：スタック）
・待ち行列（queue：キュー）
・リスト（list）
・木（tree：ツリー）
・グラフ（graph）

この章ではこれらのデータ構造を扱ったアルゴリズムについて説明する。

10-1　リストの操作 初級

Scratch のリストは要素の追加や削除が簡単に行える。リストの作成や要素の追加・削除などの操作について説明する。

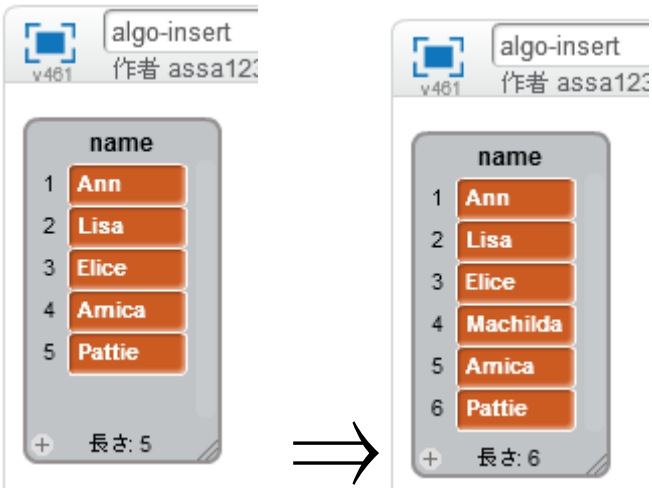
■リストの作成

プログラムの開始で、リスト name にすでにできている要素があればすべて削除し空状態に初期化する。「お名前は」という問いに名前を入力し、リスト name に追加していく。

■リストへの追加

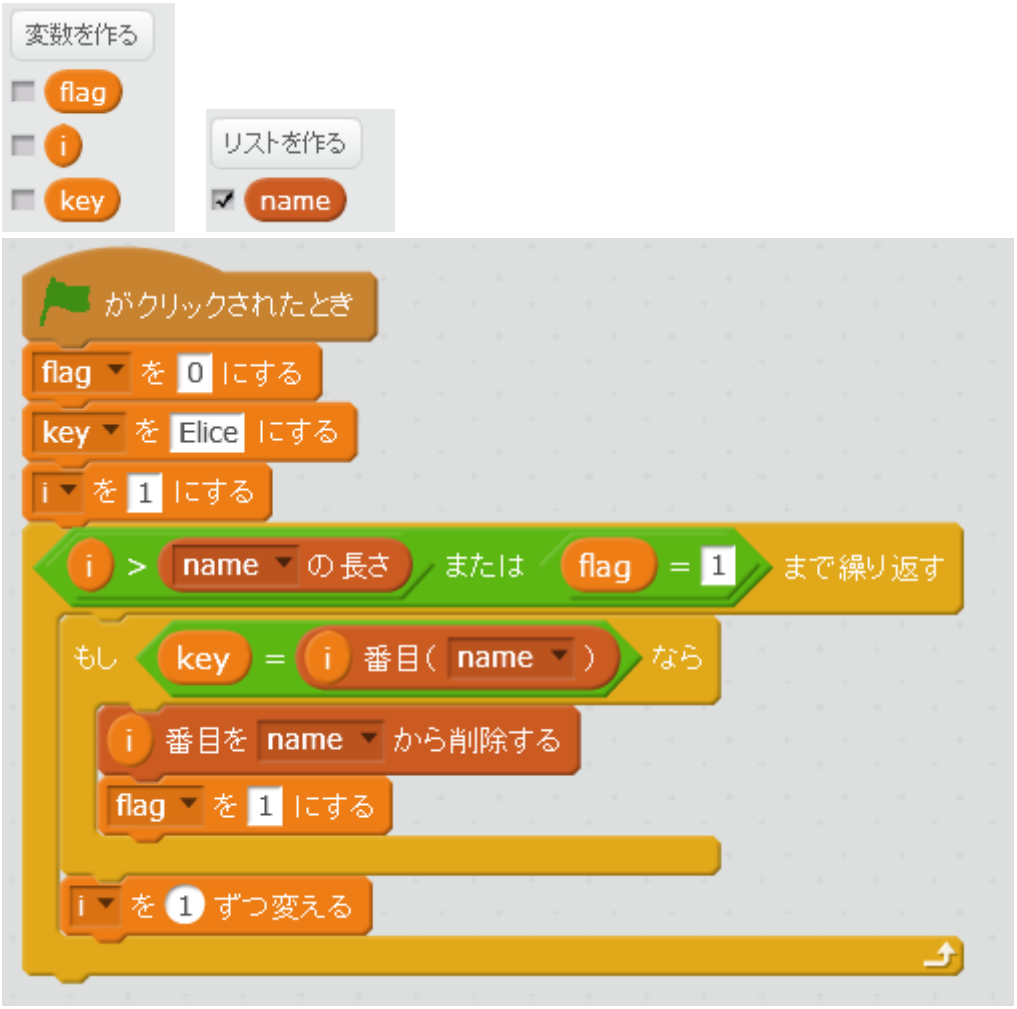
リスト name から key で示すデータを捜し、みつかったらその直後に ins で示すデータを追加する。

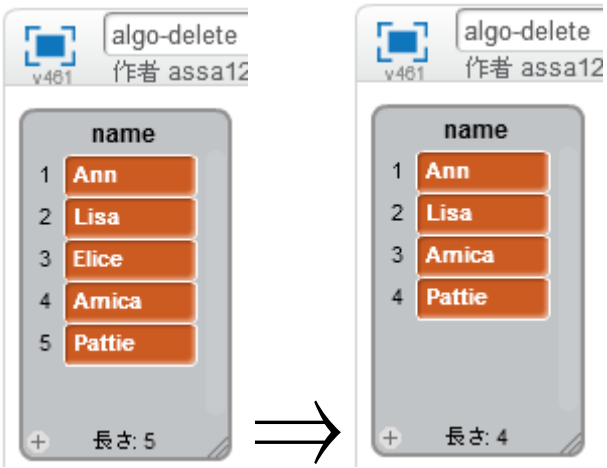

がクリックされたとき
flag を 0 にする
key を Elice にする
ins を Machilda にする
i を 1 にする
i > name の長さ または flag = 1 まで繰り返す
もし key = i 番目(name) なら
ins を i + 1 番目に挿入する(name)
flag を 1 にする
i を 1 ずつ変える


algo-insert
v461
作者 assa123
name
1 Ann
2 Lisa
3 Elice
4 Amica
5 Pattie
長さ: 5
algo-insert
v461
作者 assa123
name
1 Ann
2 Lisa
3 Elice
4 Machilda
5 Amica
6 Pattie
長さ: 6

■リストからの削除

リスト name から key で示すデータを捜し、みつかったら削除する。





11章　ゲームアルゴリズム

子ども達がプログラミングで一番作りたいものがゲームである。ゲーム機のゲームのようなものは簡単にはできないので、この章ではパズル的な簡単にできるゲームを紹介する。

11-1　サイコロ 初級

■サイコロ１つをふる

・スプライトを外部ファイルの saikoro1 にする。コスチュームとして saikoro1～saikoro6 を持つ。

・saikoro1

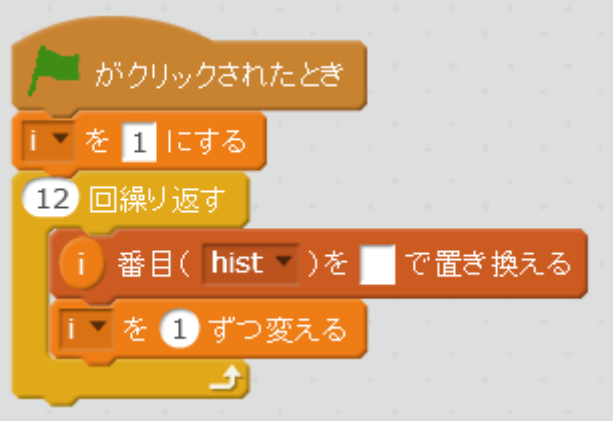
・1～6 の乱数でコスチュームを決める。
・サイコロが回っているように見せる。

■サイコロ２つをふる

２つのサイコロの目の和をリスト hist にカウントする。

・スプライトを Splite1 と外部ファイルの saikoro1 にする。

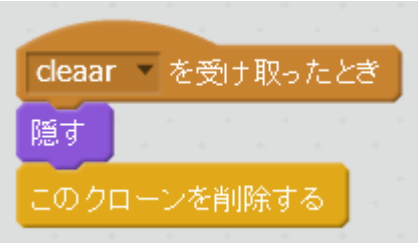
・Splite1

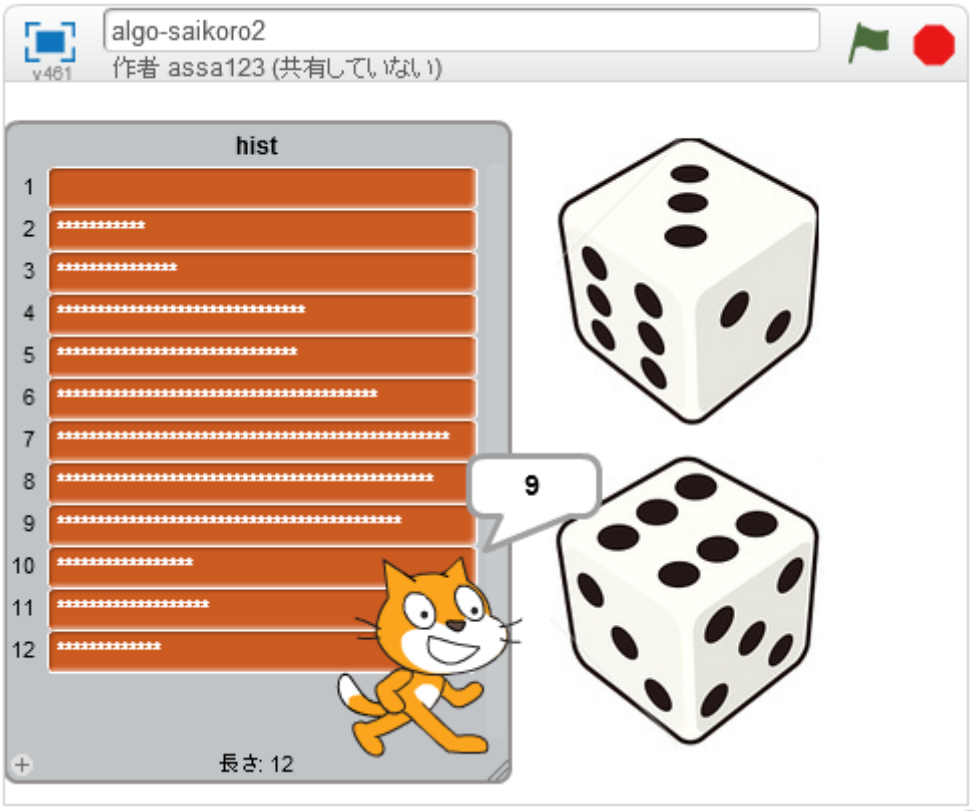

がクリックされたとき
i を 1 にする
12 回繰り返す
i 番目(hist)を で置き換える
i を 1 ずつ変える


このスプライトがクリックされたとき
cleaar を送る
y を -80 にする
a を 1 から 6 までの乱数 にする
s を a にする
saikoro を送る
0.6 秒待つ
y を 80 にする
a を 1 から 6 までの乱数 にする
s を s + a にする
saikoro を送る
0.6 秒待つ
s と言う
s 番目(hist)を s 番目(hist) と * で置き換える

・saikoro1


saikoro を受け取ったとき
自分自身 のクローンを作る
表示する
コスチュームを saikoro と a にする
x座標を 100 、y座標を y にする
3 回繰り返す
20 から 60 までの乱数 度回す
5 から 10 までの乱数 歩動かす
0.1 秒待つ
90 度に向ける
x座標を 100 、y座標を y にする


cleaar を受け取ったとき
隠す
このクローンを削除する


algo-saikoro2
v461
作者 assa123 (共有していない)
hist
1
2
3
4
5
6
7
8
9
10
11
12
長さ: 12
9

12章　ミスレニアス・アルゴリズム

今までの各章で説明したアルゴリズムのカテゴリに属さない雑多なアルゴリズムを紹介する。ミスレニアス（miscellaneous）は雑多の意。

12-1　万年歴
12-2　電卓
12-3　３拓クイズ
12-4　お絵描きツール
12-5　足し算の筆算
12-6　ローマ字入力（ひらがな）

12-1　万年歴 中級

　指定した西暦と月のカレンダーを表示する。
グレゴリオ暦では以下の規則で閏年を設けます。
①西暦の年数が 4 で割り切れる年を閏年とする。
②西暦の年数が 100 で割り切れる年は閏年からはずす。
③西暦の年数が 400 で割り切れる年は閏年に戻す。

プログラムの主なポイントは以下である。
・ｙ年が閏年かの判定で 2 月の日数を設定する
先に示した①~③の規則から西暦ｙ年が閏年である条件は、「y が 4 で割り切れかつ 100 でわりきれない」または「ｙが 400 で割り切れる」となる。

・ｙ年 m 月 1 日の曜日を調べる
　グレゴリオ暦は 1582 年に導入されたので、それ以前の年に対しては意味をもたないが、計算の都合上、西暦１年１月１日を月曜日とする。
　365 日を 1 週間の 7 で割れば 1 余るので、閏年がなければ、翌年の１月１日の曜日は今年の１月１日の曜日の次になる。従って、もし閏年がないとしたら、西暦ｙ年m月１日の曜日は日曜日からｙ日分ズレることになる。しかし実際には①~③の規則に従って閏年が入っているので、実際のズレ bias は以下のように表せる。閏年の補正は y-1 年分が対象となる。
　bias=Int(y)+Int((y-1)/4)-Int((y-1)/100)+Int((y-1)/400)

また、ｙ年の１月～(m-1)月までの合計日数は
　s=month[1]+ month [2]+・・・+ month [m-1]

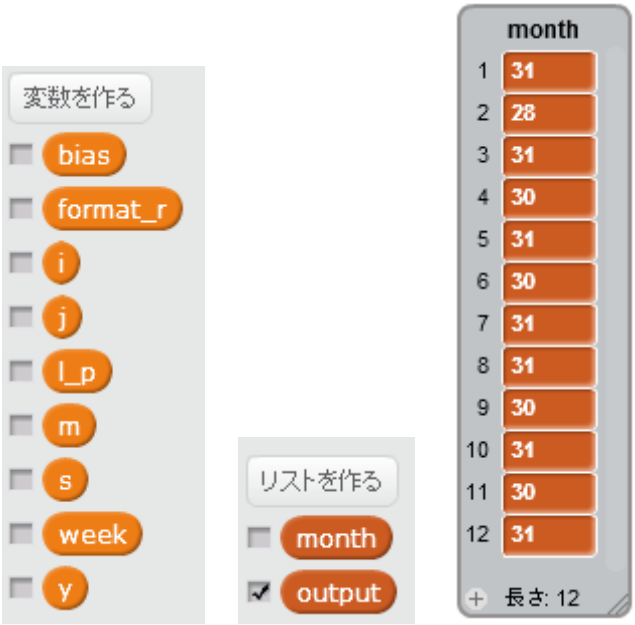
となるので、結局ｙ年m月１日の曜日は
　week=(bias+s)%7

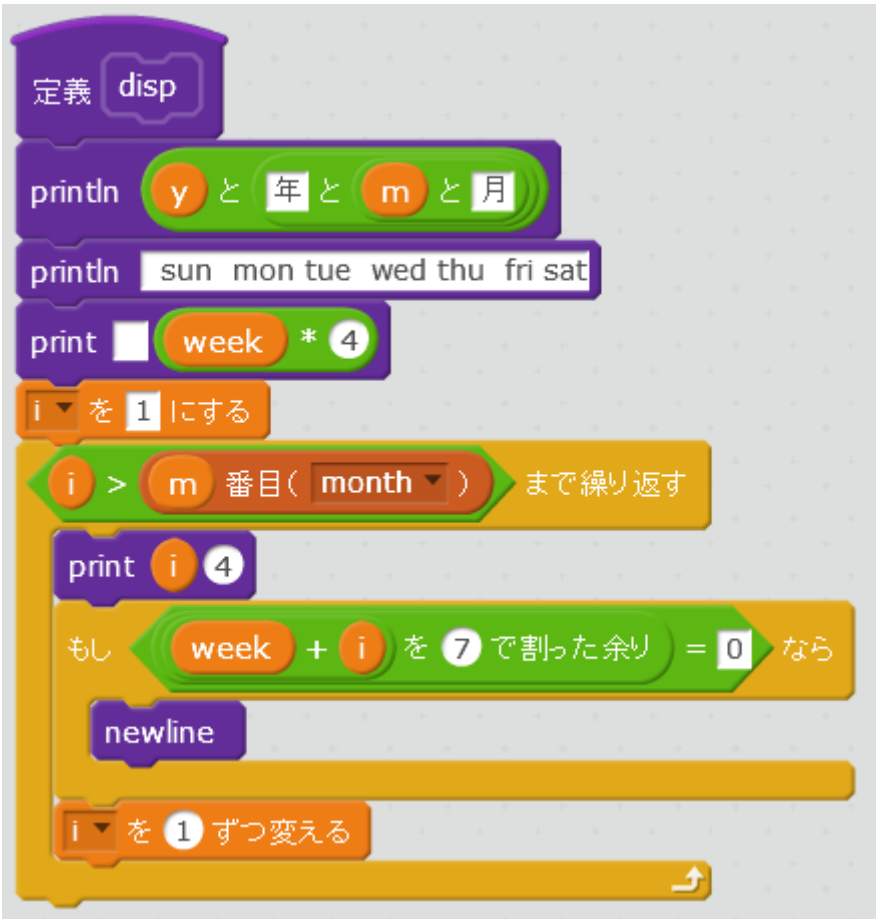
であらわせ、week が 0 なら日曜日、・・・6 なら土曜日と判定できる。

■万年歴をリストに表示

・y 年 m 月のカレンダーを、リスト output に表示する。

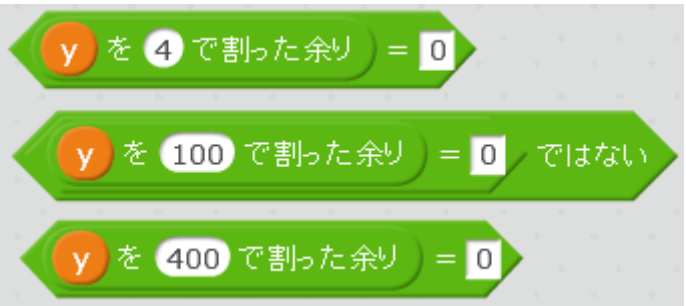
・月の先頭の 1 日の表示位置は week*4 個の空白をとった位置とする。






がクリックされたとき
init
y を 2018 にする
m を 8 にする
もし かつ または なら
2 番目(month)を 29 で置き換える
でなければ
2 番目(month)を 28 で置き換える
bias を y + - + にする
s を 0 にする
i を 1 にする
i > m - 1 まで繰り返す
s を s + i 番目(month) にする
i を 1 ずつ変える
week を bias + s を 7 で割った余り にする
disp

・赤枠には以下が入る。


y を 4 で割った余り = 0
y を 100 で割った余り = 0 ではない
y を 400 で割った余り = 0

・青枠には以下が入る

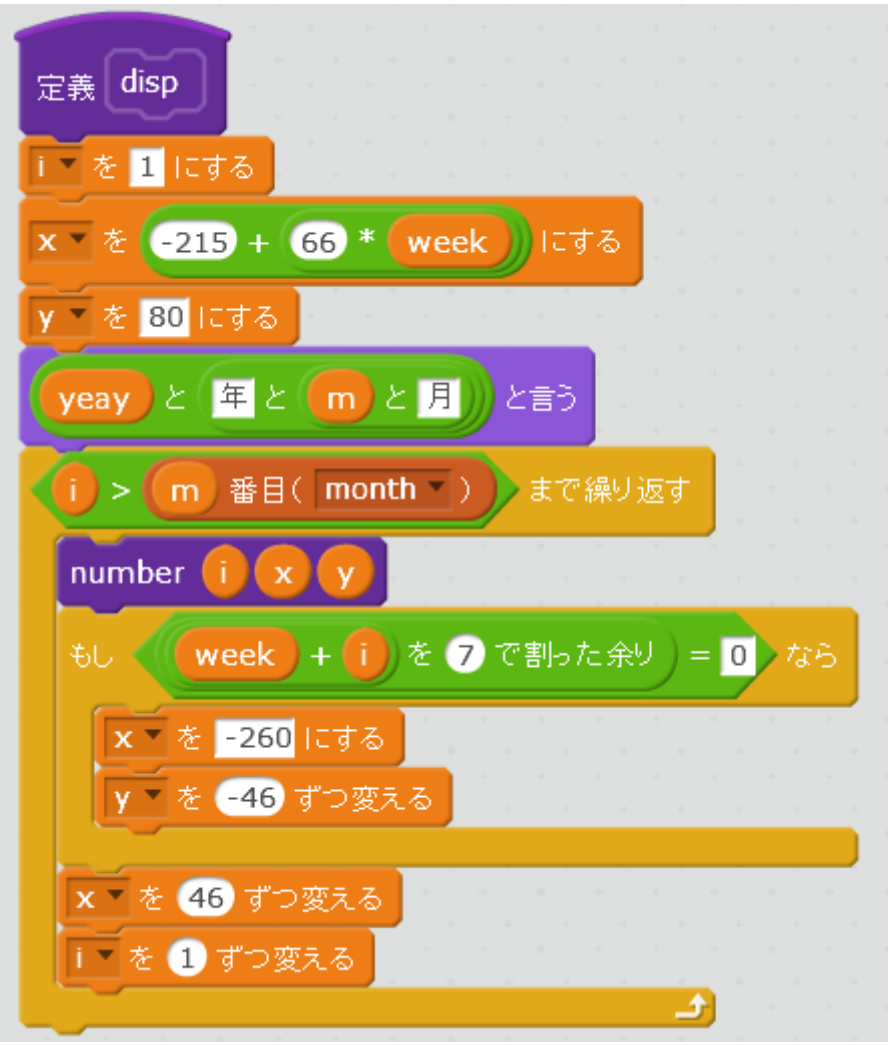

切り下げ (y - 1 / 4)
切り下げ (y - 1 / 100)
切り下げ (y - 1 / 400)

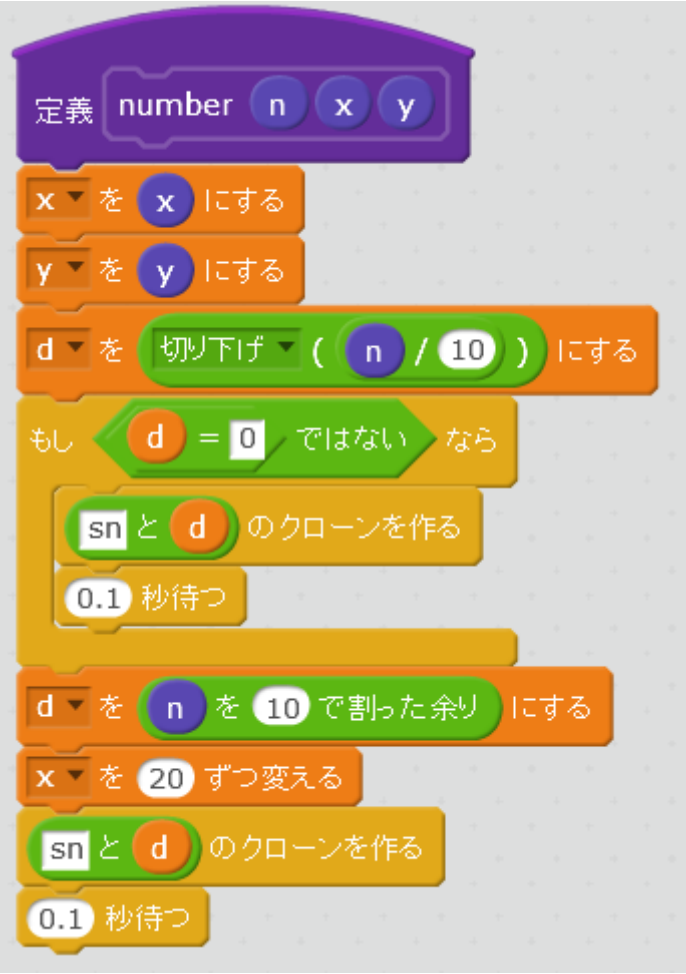

algo-mannen1
作者 assa123 (共有中)
output
1 2018年8月
2 sun mon tue wed thu fri sat
3 1 2 3 4
4 5 6 7 8 9 10 11
5 12 13 14 15 16 17 18
6 19 20 21 22 23 24 25
7 26 27 28 29 30 31
8
9
10
11
12
13
14
長さ 14

■万年歴を数字イメージを使って表示

・スプライトを外部ファイルの calendar,sn0～sn9 とする。

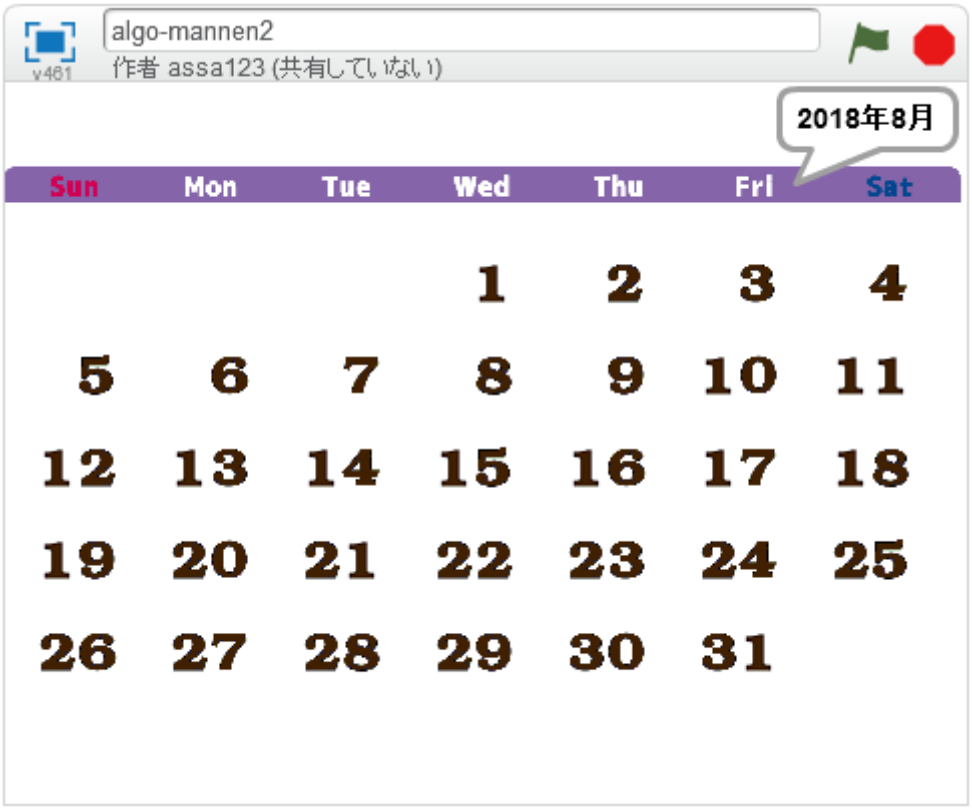
・calendar


定義 disp
i を 1 にする
x を -215 + 66 * week にする
y を 80 にする
yeay と 年 と m と 月 と言う
i > m 番目(month) まで繰り返す
number i x y
もし week + i を 7 で割った余り = 0 なら
x を -260 にする
y を -46 ずつ変える
x を 46 ずつ変える
i を 1 ずつ変える


定義 number n x y
x を x にする
y を y にする
d を 切り下げ (n / 10) にする
もし d = 0 ではない なら
sn と d のクローンを作る
0.1 秒待つ
d を n を 10 で割った余り にする
x を 20 ずつ変える
sn と d のクローンを作る
0.1 秒待つ

・sn0～sn9

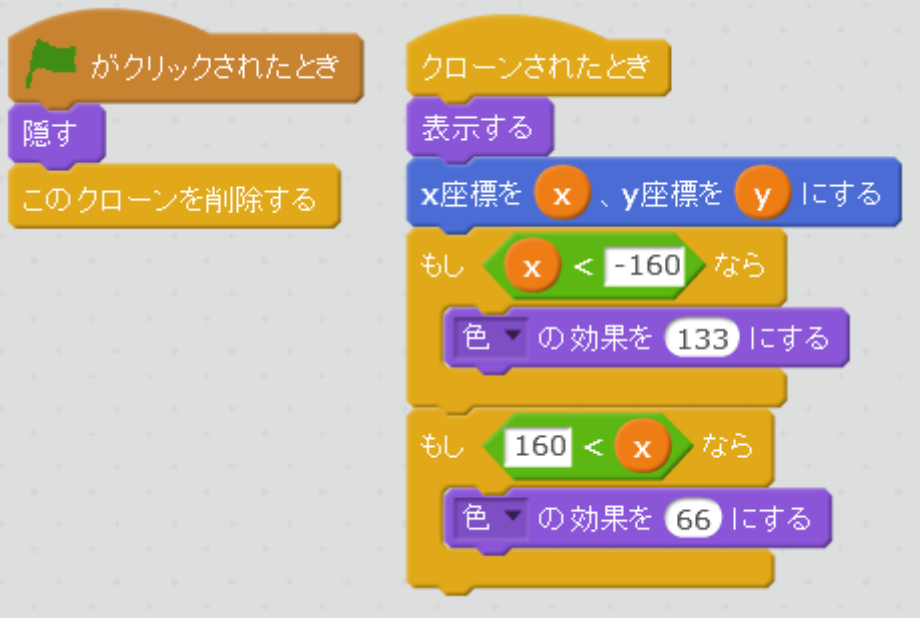

がクリックされたとき
隠す
このクローンを削除する
クローンされたとき
表示する
x座標を x 、y座標を y にする

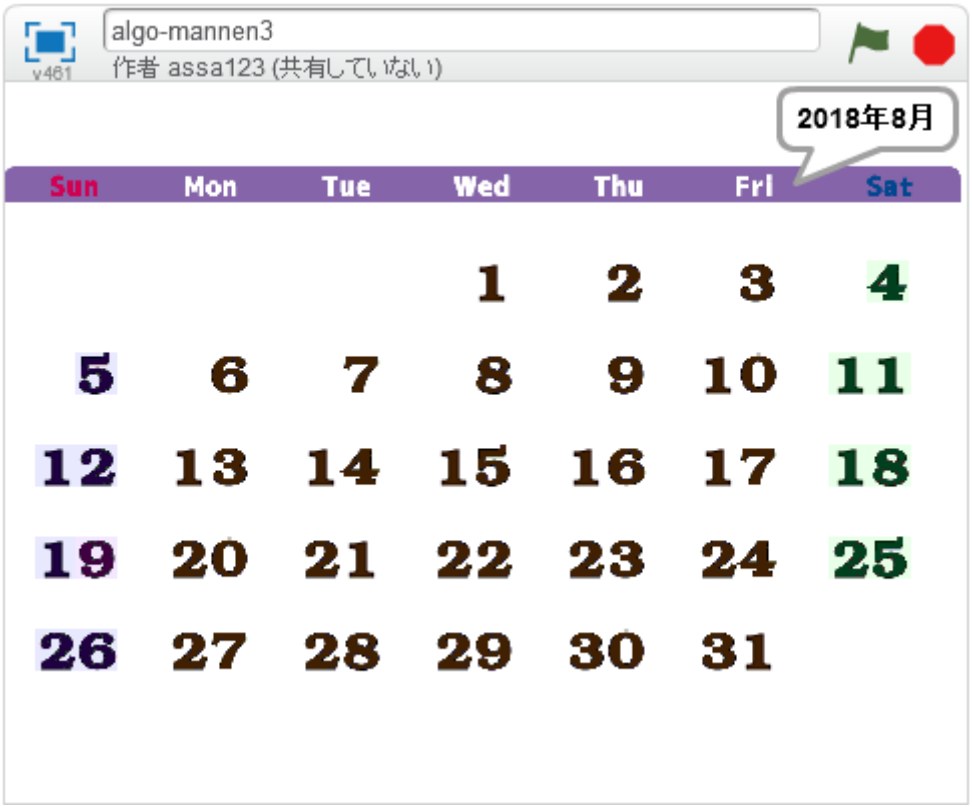

v461
algo-mannen2
作者 assa123 (共有していない)
2018年8月
Sun
Mon
Tue
Wed
Thu
Fri
Sat
1 2 3 4
5 6 7 8 9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31

■土日の色を変える

「注」白黒イメージはは余り色の変化がない。

・sn0～sn9





Scratch アルゴリズム辞典

2018 年 8 月 1 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．ソフトウエアラボ
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