

Processing

第 6 回



松田小学校／寄小学校

6-ステップ 0 : 前回の復習だよ

下のコードを打って、図形を動かしてみよう。

```
1 int a=1;
2
3 void setup() {
4   size(800, 800);
5 }
6
7 void draw() {
8   background(100);
9   
10  ellipse(a, 100, 80, 80);
11 }
```

図形を動かすには、10行目の変数 a を増減させる式を、9行目に入ればOKだよ。

a が 1 ずつ増えていく式だ。

上のコードを、図形が縦に動くように変えてみよう。

上のコードを元にして、図形が右下から左上に動くように変えてみよう。

```
1 int a=800;
2 int b=800;
3
4 void setup() {
5   size(800, 800);
6 }
7
8 void draw() {
9   background(100);
10  
11  
12  ellipse(a, b, 80, 80);
13 }
```

a が 1 ずつ減っていく式だ。

b が 1 ずつ減っていく式だ。

上のコードを、図形がウィンドウの中央で止まるように変えてみよう。

```
1 int a=800;
2 int b=800;
3
4 void setup() {
5   size(800, 800);
6 }
7
8 void draw() {
9   background(100);
10  a=a-1;
11  b=b-1;
12  if (a<=width/2) a=width/2;
13  
14  ellipse(a, b, 80, 80);
15 }
```

a がウィンドウの中央で止まる式だ。

b がウィンドウの中央で止まる式だ。

12行目をコピーして、手直すんだよ。

- ① 12行目の先頭に // を付けて実行してみよう。
- ② ①で付けた // を消して、13行目に // を付けて実行してみよう。

ball-stop で保存しよう。

6 ステップ 1 : サイコロを振ってみよう = 乱数だ

サイコロを振ったことがあるかな？ サイコロには1～6まで数字が書いてあるけど、どれが出るかは不規則で、運任せ、行き当たりばったりだよ。行き当たりばったりで数字を発生させる命令を、**random(ランダム)** と言うんだ。そして、`random();` が発生させる数字を乱数と呼ぶんだよ。

`random` は () の中に入れた数字の間で乱数を発生させ、`random(0, 700);` といえば0～699の数字を発生させるよ。() の中に (1, 6) と書くこともでき、その場合は、1～5までの数字だ。ただし、1～5と言っても、1, 2, 3, 4, 5 だけではない。1～5の間には小数たとえば、1.2や4.538 などもあるから、整数にしたい場合は工夫が必要だ。

新規を開いてコードを打って実行してみよう。

```
1. println(random(1, 6));
```

コンソールにでた数字は 3.7703238 だったよ。みんなのパソコンででた数字は全部違うはずだ。隣のトモちゃんを確認し合ってみよう。

また、サイコロに使うには、整数で6もでないと困るね。そこで下のように変えるんだ。今度はいくつができる？

```
1. println(int(random(1, 7)));
```



それぞれ新規を開いて、下のコードを打って `random();` を使ってみよう。

```
1 float x=random(600);
2 float y=random(1, 400);
3
4 void setup() { コピー&貼り付け
5   size(600, 400);
6 }
7
8 void draw() {
9   fill(255, 0, 0);
10  ellipse(x, y, 200, 200);
11 }
12
```

コピー&貼り付け

上のコードは、実行するたびに円の場所が違うね。

```
1 float r=random(256);
2 float g=random(256);
3 float b=random(256);
4
5
6 void setup(){
7   size(600, 400);
8 }
9
10 void draw(){
11  background(100);
12  fill(r, g, b);
13  ellipse(300, 200, 300, 300);
14 }
15
```

上のコードは、実行するたびに円の色が違うね。

ミッション

上の2つのコードを合体しよう。同じコードは書かなくても良いね。変数 `float` は内容が違うから、コピペで全部書くんだね。

`fill` と `ellipse` は、左と右では内容が違うから全部必要だ。しかし、コードを打たないで、コピー&貼り付けを使うんだ。左の9行目と10行目を、右のコードの一番下にコピペしよう。

できたら `random` で保存しよう。

6-ステップ 2 : random=乱数の続き

random と for文(繰り返し)を組み合わせると、さまざまに楽しいことができるよ。下のコードは、線を100本描く繰り返し命令だ。まず、線の色の r にだけrandom がかけてあるから、赤系の色が乱数で生まれるね。そして、線の始点と終点のX座標が乱数だから、実行されるたびにX座標が行き当たりばつりに動くわけだね。

ファイルから新規を開いて下のコードを打ってみよう。

```

1 size( 700, 700 ); ← line の700 と同じことがキモ
2 background( 255 );
3
4 for( int i = 0; i < 100; i=i+1 ) {
5   float a=random(0,255);
6   stroke(a,100,100); ← stroke(r,b,g) の r を変数に
7   line( random( 700 ), 0, random( 700 ), 700 );
8 }
9

```

参照 : コンピューターの基礎Ⅱ <https://cc.musabi.ac.jp/kenkyu/cf/renew/program/processing/processing10.html>



ミッション

- ① stroke の他の色にも r と同じ random をかけてみよう。ここで実行すると、どう変わったかな？
- ② line のY座標の始点と終点も、X座標と同じ random をかけてみよう。実行すると、どうなるかな？
- ③ line を ellipse に変えて実行するとどうなるかな？

できたら random-1 で保存しよう。

新しいファイルに、下のコードを打って実行してみよう。(ファイル→新規)

```

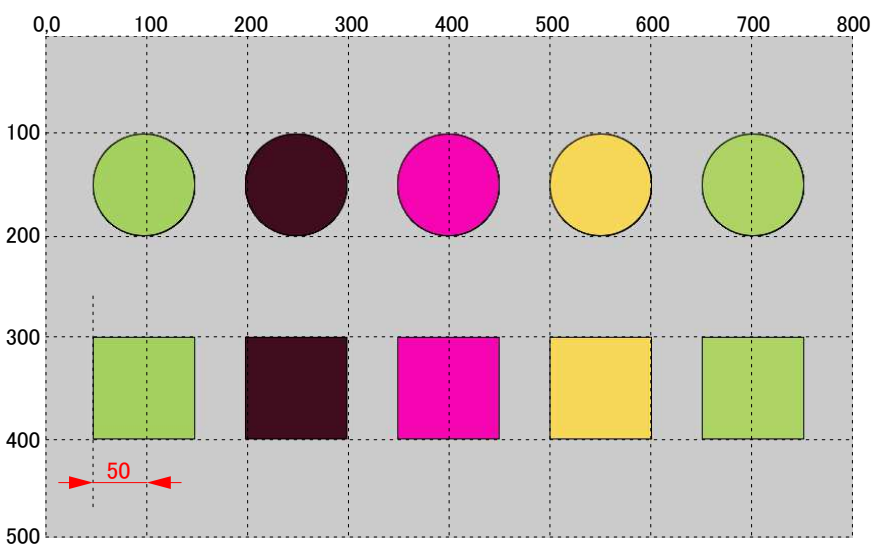
1 void setup() {
2   size(800,500);
3   frameRate(5);
4 }
5
6 void draw() {
7   for(int i=100; i<=750; i=i+150) {
8     fill(random(0,255), random(0,255), random(0,255));
9     ellipse(i, 150, 100, 100); ← コピペだよ
10    rect(i-50, 300, 100, 100);
11  }

```

左のコードを実行すると、下のような結果ができたかな？

fill(r,g,b): にそれぞれ random(0,255) を入れて、テンデコノの数字を発生させて、色決めをしている。それを、for 文と組み合わせて、動かしてみたんだ。

円や四角形のX座標に i が入るたびに、テンデコノの数字が入るから、そのたびに色が違っているんだね。



円と四角形の色が揃っているのを見ると、第1回目に 7→10行目と実行し、第2回目に再び 7→10行目と実行して、第3回目に入って繰り返しているのがわかるね。

左のコードの10行目に注目して。X座標が i-50 となっているのはなぜだろう？

もし、-50 しないと、四角形はX座標が 100 から始まってしまうんだね。円の中心点と四角形の基準点の違いをそろえるために、i-50 としたんだ。

random-2 で保存しよう。

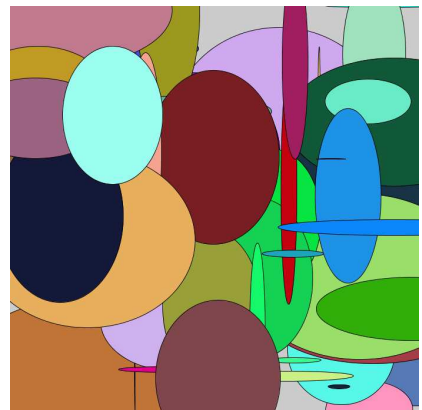
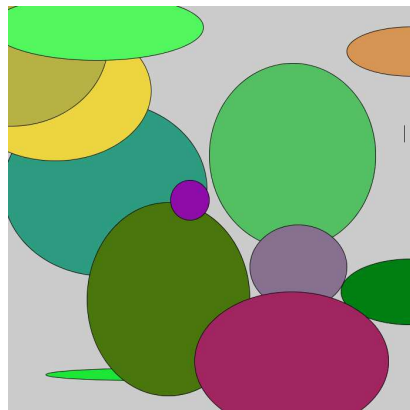
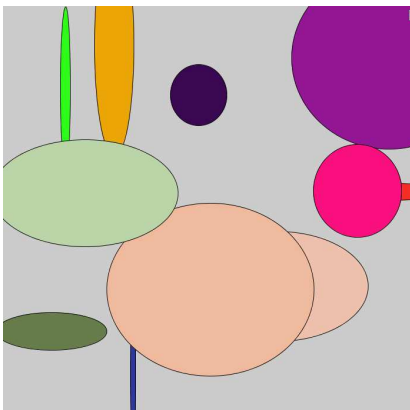
6-ステップ 3 : random=乱数を深めてみよう

新しいファイルに、下のコードを打って実行してみよう。

```
1 void setup() {  
2   size(700, 700);  
3 }  
4  
5 void draw() {  
6   fill(255, 0, 0);  
7   ellipse(200, 500, 100, 100);  
8 }
```

ミッション-1

左のコードは動かすように viod を使っているけど、これでは動かないね。このコードに数行をくわえて、random に動かしてみよう。まず前回やったように、色を random にかえてみよう。6行目の r, g, b を random(0, 256) に変えてみるとどうなるかな？ 次に ellipse のX座標とY座標に random(,) を使ってみると、どんな感じになる？ random(,); の数字は各自で考えてみてね。



左から右のように、めまぐるしく楕円が生まれてきたね。これが random のパワーだ。

ミッション-2

7・8行目の2行変えただけでも、ずいぶん変わったね。でも、円の周囲が何だかすっきりしない。そこで縁取りをなくす命令を size の下に入れてみよう。縁取りをなく命令を覚えているかな？ noStroke(); だったね。大文字に気をつけて。

ずいぶんと激しく変化するね。これは1秒間に60回実行しているからなんだ。そこで実行回数をコントロールしてみよう。それには、frameRate(); という命令を使えば良かったんだ。書き加える場所を考えてね。カッコの中に入れる数字によって、回数が変わるから試して。どんな命令でも () の中の数字を変えると、動きが大きく変わるから実験してみよう。

背景の色も変えてみよう。背景の命令は覚えているよね。忘れていたら、となりのトモちゃんに助けを求めよう。無彩色なら () の数字は？ また r, g, b にどんな数字を使えば良いか？ いろいろ試してみよう。

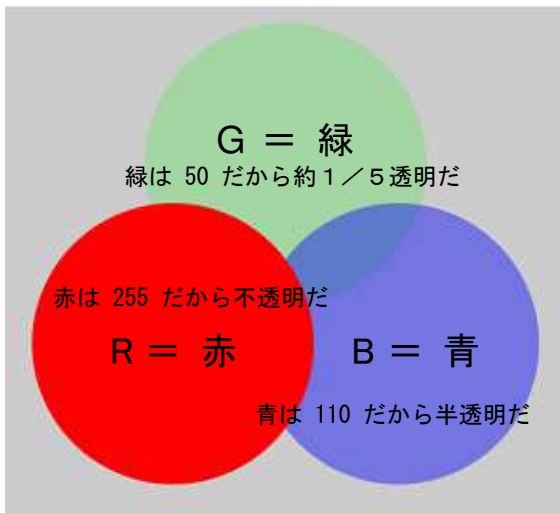
ミッション-3

最後だ。円の大きさにも random を入れたらどうなるだろう？ X方向、またはY方向だけ random にするとどうなる？ 両方とも random にするとどう変わる？ そして、size(700, 700); を fullScreen(); に変えると、ナント、ナント…！

fullScreen から戻るには、左上隅の Esc を押そう。戻らなくなったら、どこかでマウスを空クリックさせてから Esc を押してね。

random-3 で保存しよう。

6-ステップ 4 : 再度、色を攻めてみよう



```
1 size(500, 500);   ファイル→新規
2 noStroke();      コード打って実行だ
3
4 fill(0, 255, 0, 50);
5 ellipse(250, 140, 250, 250);
6                 上の2行をコピー、貼り付け、そして直しだ
7 fill(0, 0, 255, 110);
8 ellipse(350, 300, 250, 250);
9                 2行をコピー、貼り付け、そして直しだ
10 fill(255, 0, 0, 255);
11 ellipse(150, 300, 250, 250);
12
```

このコードはすでに学んだ RGB を使っている。でも、fill の () の中には、4つの数字が入っているね。4番目の数字は、透明度を表しているんだ。4番目の数字も、0 から 255 の整数で、0 は完全に透明、255 は完全に不透明となるよ。

ミッション 数字を変えて確かめてみよう

ファイルから 新規を開いてコードを打って実行してみよう。

```
1 float a=1;
2 float b=1;
3
4 void setup(){
5   size(1500, 500);
6   noStroke();
7   frameRate(20);
8 }
9
10 void draw(){
11   background(255);
12   a=a+2;
13   fill(255, 0, 0, a);
14   b=b+10;
15   ellipse(b, 300, 200, 200);
16 }
17
```

透明度を変える役割

左のコードはすべて以前に学んだものばかりだ。14、15行目のコードは、円を左から右へ移動させるコードで第5回で学んだね。たった1つ、13行目のfill の () の中に第4の数字が登場したことが新しい。

今回加えた fill の a は透明度を司る数字で、最初に float a=1; と宣言してある。だから小数で、しかも最初は透明で登場するはずだ。そして、**a=a+2;** によって少しずつ透明度を減らしていき、右端に到達する最後には 255 となって、まったく不透明の真っ赤になるんだ。

透明度を操作するだけでも、幻想的なイメージが創れそうだね。いろいろとコードや数字を変えて、試してみよう。



左から薄い赤色で登場し、右へと移動しながら赤色が増していくね。これも透明度を操作しているからできるんだ。if 文を使えば、跳ね返ってきたり、往復させたりもできるよね。

いろいろと試してみたら、toumei で保存してね。

6-ステップ 5 : 別の角度で、色を攻めてみよう

ファイルから 新規を開いてコードを打って実行してみよう。

```
1 float x = 300;
2 float y = 200;
3 float r = 100;
4
5 void setup(){
6 size(600, 400);
7 }
8
9 void draw(){
10 background(255);
11
12 x = x + random(-5, 5);
13 y = y + random(-3, 3);
14 r = r + random(-2, 2);
15
16 strokeWeight(20);
17 stroke(0, 0, 255);
18 ellipse(x, y, r, r);
19 }
20
```

コピー、貼り付け、
そして直した

コピー、貼り付け、
そして直した

255 の後ろに random(255) を加える

ミッション

左のコードは、random を使って揺れを描いているんだ。透明度の制御も学んだので、青色の円に透明度を加えてみよう。透明度も random を使うと、不思議な効果が出てくるよ。

17行目の `stroke(0, 0, 255);` の第4番目の数字として、`random(255)` を加えて実行してみよう。

色を表すのは、R,G,B だけじゃない。新規ファイルで、下のコードを打って実行してね。

```
1 size(400, 150);
2 colorMode(HSB, 360, 100, 100, 100);
3
4 noStroke();
5 fill(230, 80, 80, 50);
6 rect(100, 50, 200, 50);
7
8 fill(200, 60, 50, 30);
9 ellipse(100, 75, 100, 100);
```

HSB宣言

この色表現はHSBといって、これを使うには、最初に `colorMode(HSB,360,100,100,100);` と宣言する必要があるんだ。何も宣言をしなければ RGB モードが使われるよ。
`colorMode(HSB,360,100,100,100);`の宣言をするとこの宣言から下は**HSBモード**になって、柔らかい色合いにできるよ。ただし、**Hは360、SとBは100、最後の透明度も100**までの数字で指定するんだ。

Hは色相(Hue)、Sは彩度(Saturation)、Bは明度(Brightness)、第4の数字は透明度(Transparency)をあらわすんだけど、こんなことは覚えなくても良いよ。

ツールから色選択を選んで、カラーセレクトターをだしてみよう。



色決めには番号が必要だね。
RGB ではカラーセレクトターで欲しい番号を拾ったけど、HSB も同じことが出来るよ。左の図の HSB の番号を拾えば良いんだ。

それともう1つ、一番下の白い窓に `#E01010` って出ているね。`#E01010` を `fill` などの `()` の中に入れてやれば、これでも同じ色を表すことができる。

数字から色のイメージが掴めないけど、将来ホームページを作る時には大活躍するよ。

6 ステップ 6 : 色は楽しい

ファイルから 新規を開いてコードを打って実行してみよう。

```
1 void setup(){
2   size( 500, 500 );
3 }
4
5 void draw(){
6   int r = mouseX; // マウスのX座標
7   int g = mouseY; // マウスのY座標
8   int b = 100;
9   background( r, g, b );
10 }
11
```

hsb で保存しよう。

左のコードは、たった7行しかないね！しかし、`mouseX` と `mouseY` という新しいコードがでている。これは、マウスのカーソルの置かれた位置が、X座標であり、Y座標だという意味だ。だからマウスを動かすと、カーソルの位置に従って、X座標とY座標が動いていくんだ。

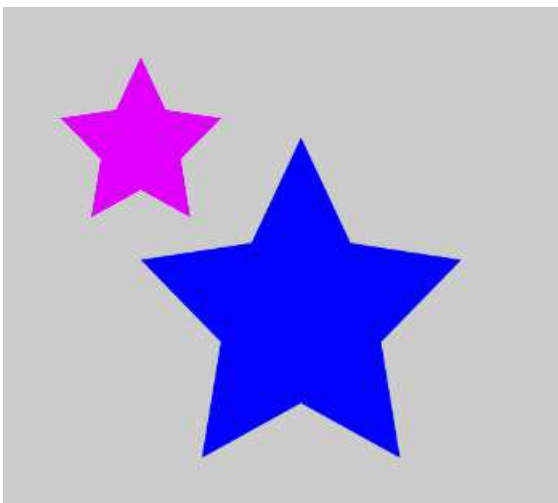
では実行してみよう。最初は色付きの500*500 のウィンドウが表れるだけだ。しか～し、カーソルをウィンドウの上に置くと不思議なことがおきるね。

第5回のステップ1で、`mousePressed` っていうコードをやったよね。マウスを押したらっていう意味だった。今回はマウスを置いただけでも反応してくれる。

マウスだけではなくキーにも役割を持たせることができるけど、それは次回にしよう。コンピューターそしてプロセッシングには、まだまだいろいろな働きを持たせることができるんだ。楽しみにしててほしい。

次回の予告

今まではさまざまな図形を操作してきたけど、画像にはまだ手を付けていなかった。画像や写真を取り込むには、ちょっとした決まり事があるんだ。次回は、画像や写真を取り込んでいじってみよう。



星形は作図することもできるね。でも、ネコを作図するのは難しそうだ。そこで画像を取り込むことになるんだ。

Bye-bye!