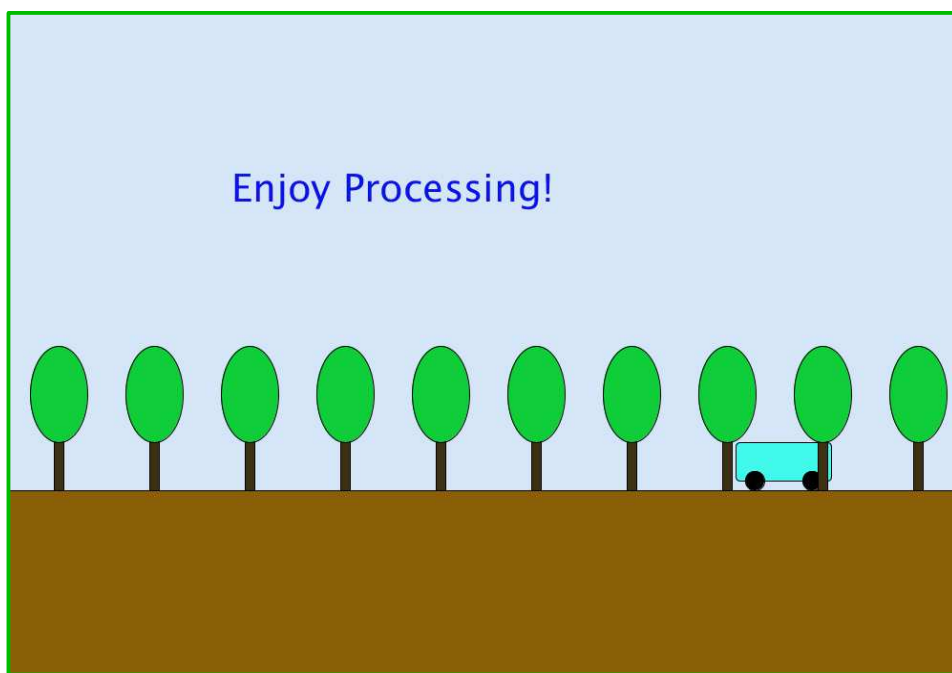


# Processing

## 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、図形が左上から右下に動くように変えてみよう。

```
1 int a=1;
2 int b=1;
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 ずつ増えていく式だ。

b が 1 ずつ増えていく式だ。

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=1;
2 int b=1;
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 }
```

12行目をコピー&ペーストして、手直するんだよ。

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。

- 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-stop2 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

サイコロには1~6まで数字が書いてあるけど、どれが出るか運任せだね。運任せに数字を発生させる命令を、**random(ランダム)** といって、`random()`; が発生させる数字を乱数と呼ぶんだ。

`random` は `()` の中に入れた数字の間で乱数を発生させ、`random(0, 700)`; といえは 0~699.999 つまり 0~700未満の数字を発生させるよ。0 から発生させる場合は `random(700)`; と書いてもOKだ。

`()` の中に `(1, 6)` と書いた場合は、1~5(6未満) までの数字だ。ただし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));
```

みんなのパソコンで出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  
いくつがでる？

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

`random` の前に `int` を入れると、整数だけが使われるんだ。  
これで 1~7未満の整数、つまり1~6ということになるね。



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() {
5   size(600, 400);
6 }
7
8 void draw() {
9   fill(255, 0, 0);
10  ellipse(x, y, 200, 200);
11 }
```

コピー&ペーストして、手直し

左のコードを何回か実行すると、  
実行のたびに円の場所が変わっているね。  
円の場所を、  
X 座標は `random` で、0~600未満  
Y 座標は `random` で、1~400未満  
と指定しているから、場所が変わるんだ。

できたら `random` で保存しよう。

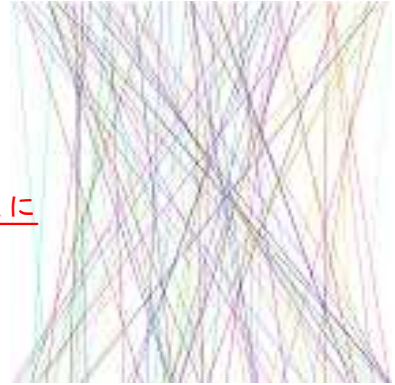
## 6-ステップ 2 : random=乱数の続き

random と for文(繰り返し)を組み合わせよう。下のコードは、線を100本描く命令だ。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んだ。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座標が運任せに動くわけだね。

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   int a=int(random(256));
6   stroke(a, 100, 100); ← stroke(r,b,g) の r を変数 a に
7   line(random(700), 0, random(700), 700);
8 }
```

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

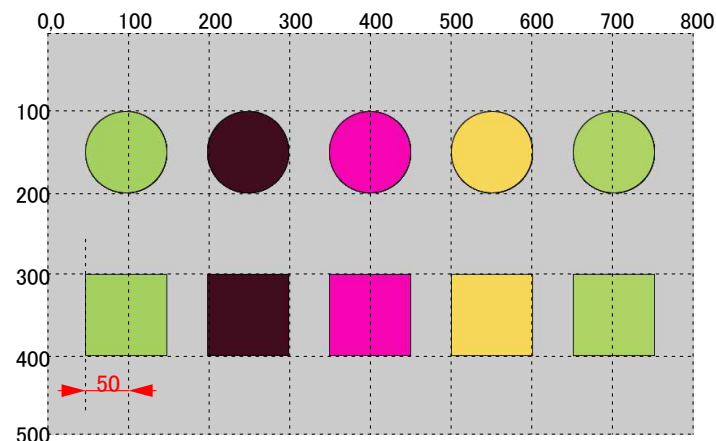
- ① stroke の他の色にも r と同じ random をかけて、実行してみよう。
- ② line のY座標の始点と終点も、X座標と同じ random をかけて、実行してみよう。
- ③ line を ellipse に変えて実行するとどうなるかな？

できたら random-1 で保存しよう。

上のコードを手直して、下のコードにして実行してみよう。

```
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(256), random(256), random(256));
9     ellipse(i, 150, 100, 100);
10    rect(i-50, 300, 100, 100);
11  }
12 }
```

コピー&ペーストだよ



fill(r,g,b); に、それぞれ random(256) を入れて、乱数で色決めをしている。それを、for 文と組み合わせた。

X座標を i-50 としないと、四角形はX座標が 100 から始まってしまうんだ。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

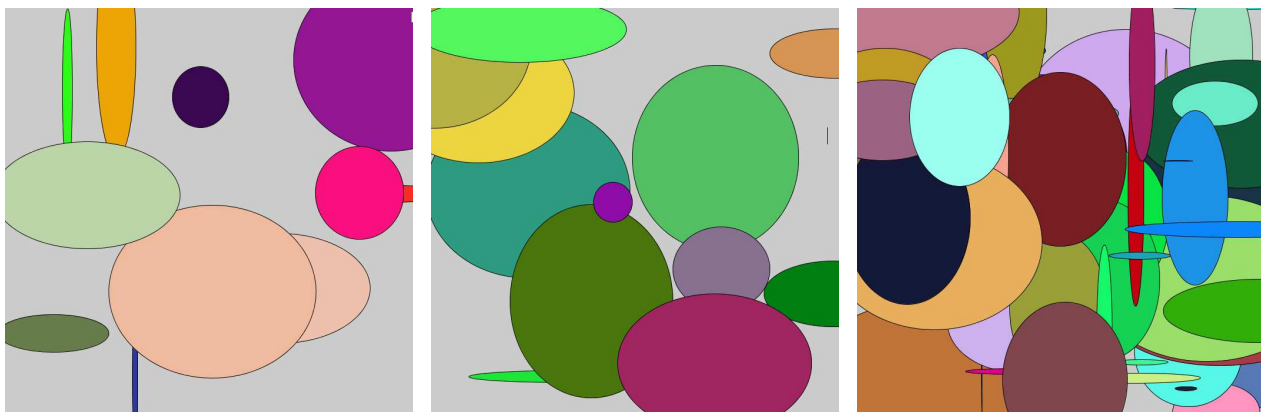
新しいファイルに、下のコードを打って実行してみよう。

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

### ミッション-1

上のコードに数行をくわえて、random に動かそう。

- ① 6行目の r, g, b を random(256)に変えるとどうなるかな？
- ② 次に ellipse のX座標とY座標にrandom( , )を使うと？
- ③ また、横直径、縦直径にrandom( , )を使うとどう変わる？  
random( , ); の数字は考えてみて。



### ミッション-2

- ① 縁取りをなくす命令 noStroke(); を size(); の下に入れてみよう。
- ② frameRate(); 命令を使って、実行回数をコントロールしよう。  
書き加える場所を考えてね。  
カッコの中に入れる数字を変えると、動きが大きく変わるから実験してみよう。

### ミッション-3

size(700, 700); を fullScreen(); に変えてみよう。

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 : random の最後

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; コピー&ペースト、
3 float r = 100; そして直しだ
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); コピー&ペースト、
14  r = r + random(-2, 2); そして直しだ
15
16  strokeWidth(20);
17  stroke(0, 0, 255);
18  ellipse(x, y, r, r);
19 }
```

### 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。青色の円に透明度 r を加えてみよう。17行目の `stroke(0, 0, 255);` の第4番目の数字として、`random(256)` を加えて実行してみよう。

255 の後ろに `, random(255)` を加える

shaking で保存しよう。

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 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 }
```

左のコードは、7行しかない！新しいコード `mouseX` と `mouseY` 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。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。では実行してみよう。色付きの500\*500のウィンドウが表れるんだけど、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。

mouse-color で保存しよう。

### 次回の予告

画像や写真を取り込んでいじってみよう。



Bye-bye!