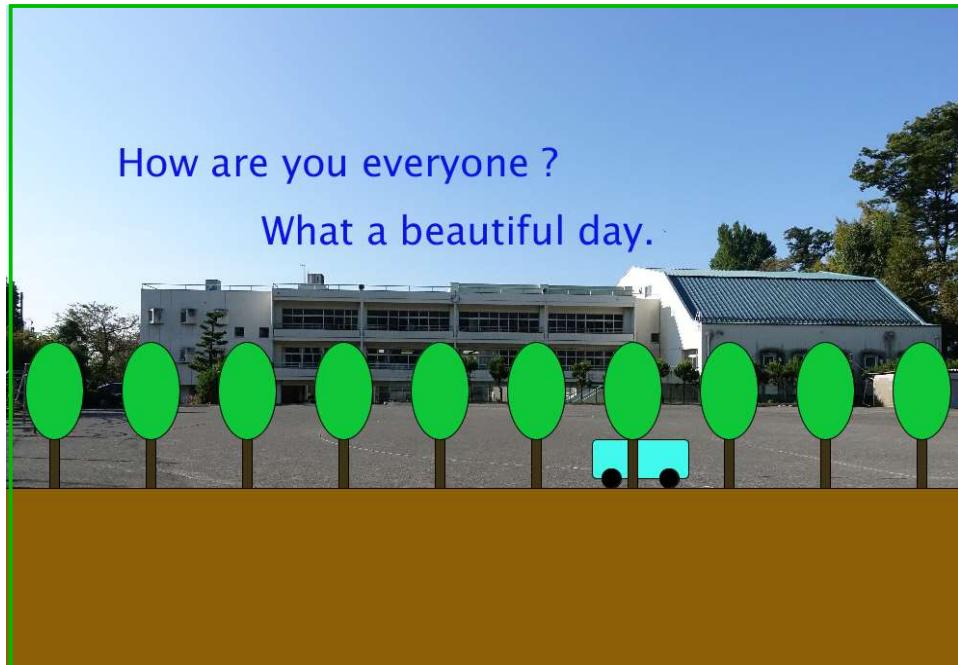


# Processing

第6回



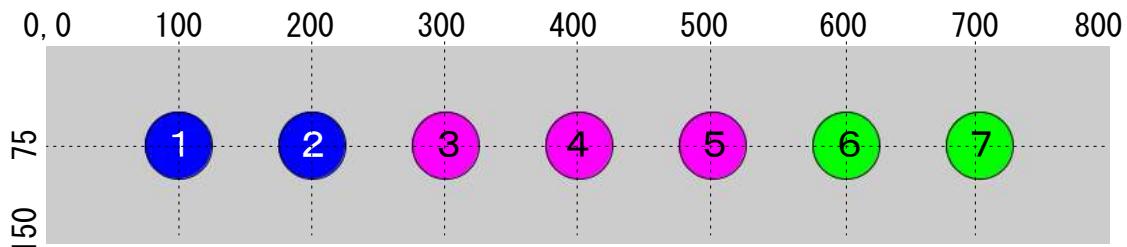
松田小学校／寄小学校

## 6-ステップ 0 : 前回の復習だよ

ファイル→新規で打ってみよう。

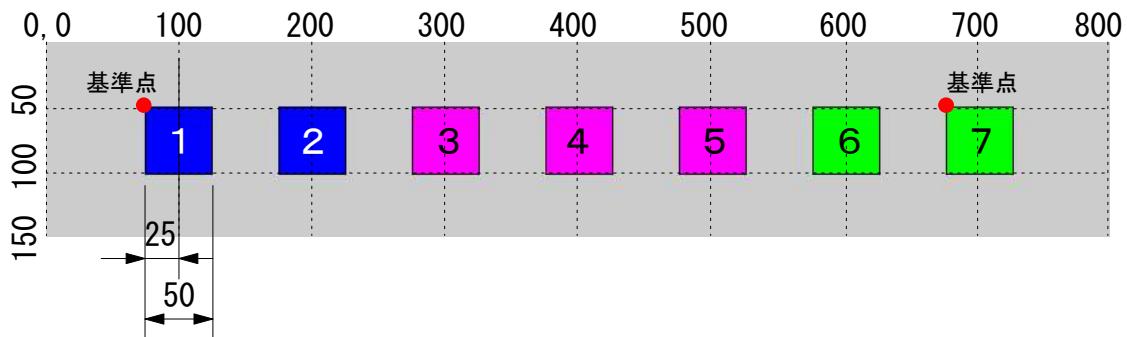
```
1 size(800, 150);
2 for (int a=1; a<=7; a=a+1) {
3   if (a<=2) {fill(0, 0, 255);}
4   if (a>=3) {fill(255, 0, 255);}
5   if (a>=6) {fill(0, 255, 0);}
6   ellipse(100*a, 75, 50, 50);
7 }
```

下図のようになったかな？



下図になるようにコードを手直ししてみよう。円は中心が基準点だったから調整は不要だった。しかし、四角形の基準点は左上の角だね。

1の基準点は、X座標が 100-25 で、Y座標は 50 だ。



**rect-7** で保存しよう。

## 6-ステップ 1：マウスを使ってみよう－動かす準備

ファイル→新規で下のコードを打ってみよう。

```
1 void setup(){
2   size(500, 300);
3 }
4
5 void draw(){
6   if(mousePressed){
7     fill(0, 0, 255);
8     rect(100, 50, 300, 200);
9   }
}
```

ウィンドウの上でマウスを押すと

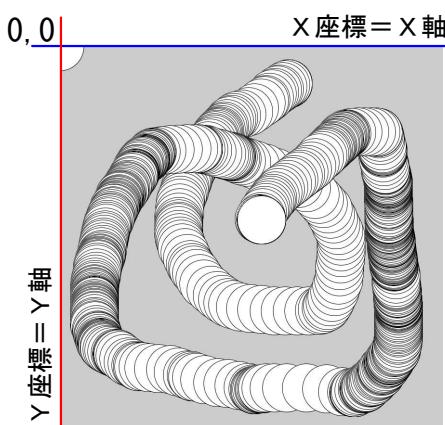
マウスが押されたら、という命令

blue で保存しよう。

ファイル→新規で下のコードを打って実行してみよう。

```
1 void setup() { ← [ ]内を最初に1度だけ実行する
2   size(800, 800);
3 }
4
5 void draw() { ← [ ]内を何度も実行をくり返す
6   circle(mouseX, mouseY, 100);
7 }
```

void setup=準備  
(「void」セッタップ)  
void draw=描く  
(「void」ドロー)



circle( mouseX , mouseY , 100 ); は、  
直径100の円の位置を、マウスのいる位置  
(mouseX は X 座標、mouseY は Y 座標) に描くという  
意味で、マウスの移動につれて図も動く。

[ ]内の円を何度も実行するので、  
直径100の円がたくさん描けた。

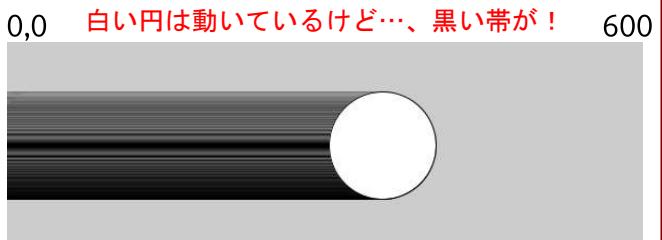
### ミッション

- ① 横にだけマウスに付いて来るようにしてみよう。
- ② 縦にだけマウスに付いて来るようにしてみよう。
- ③ 直径をmouseX, mouseY にしてみよう。 ミッション保存不要

## 6-ステップ 2：円を自動で動かそう

ファイル→新規で下のコードを打ってみよう。

```
1 int a=0;  
2  
3 void setup() {  
4     size(600, 200);  
5 }  
6  
7 void draw() { 0,1,2,3,4,5～と代入され、X座標が大きくなつて右へ動いていく。  
8     a=a+1; ←  
9     circle(a, 100, 100);  
10 }
```



### ★超重要★

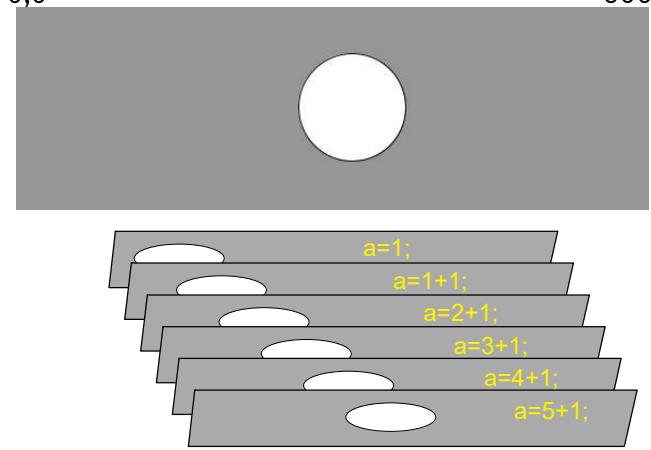
図形を動かすには、void setup(); と void draw(); が、絶対に必要なんだ。

void setup() に続く { } の中の命令を 1 回だけ実行し、その後、  
void draw() に続く { } の中の命令を何度も（1秒間に60回）実行するから、  
動いているように見えるんだ。

黒い帯を取るよう手直ししよう。

```
1 int a=0;  
2  
3 void setup() {  
4     size(600, 200);  
5 }  
6  
7 void draw() {  
8     background(150); ← 加えた  
9     a=a+1;  
10    circle(a, 100, 100);  
11 }
```

0,0 白い円がプログラムで動いている！！！ 600



ball で保存しよう。

### ミッション

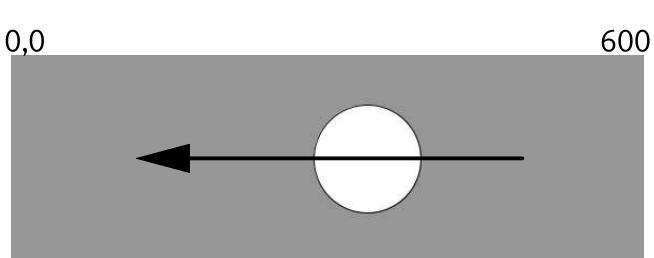
- ① 円に好きな色を付けてみよう。何行目に色のコードを書く？
- ② 円の大きさを変えてみよう。
- ③ a=a+2; にするとどう変わった？なぜだろう？

ミッションは保存不要

## 6-ステップ 3：図形の動きを制御しよう

白丸が左から表れるようにしてみよう。

```
1 int a=□; ← 初期値 (=白丸が最初にある位置)  
2  
3 void setup() {  
4     size(600, 200);  
5 }  
6  
7 void draw() {  
8     background(150);  
9     a=a □ 1; ← 1ずつ小さくなる  
10    circle(a, 100, 100);  
11 }
```



ball-1 で保存しよう。

ball を呼び出して、白丸が右に消えたら、また左から表れるようにしてみよう。

```
1 int a=0; ← aを0に戻した  
2  
3 void setup() {  
4     size(600, 200);  
5 }  
6  
7 void draw() {  
8     background(150);  
9     a=a+1;  
10    if(a>=□) {a=0;} ← 条件を加えた  
11    circle(a, 100, 100);  
12 }
```

### ヒント

白丸をまた左から表すには、もし a が 600 以上になら、a を最初の位置 (a=0) にしてやれば、また左から出てくるんだ。

ball-2 で保存しよう。

ball-2 を手直しして、左から表れた白丸を中央で止めてみよう。

```
1 int a=0;  
2  
3 void setup() {  
4     size(600, 200);  
5 }  
6  
7 void draw() {  
8     background(150);  
9     a=a+1;  
10    if(a>=□) {a=□}  
11    circle(a, 100, 100);  
12 }
```

### ヒント

白丸を止めるには、止める条件になったら、a の値を止める条件と同じ値にしてやれば、白丸は止まるんだ。たとえば、a が 300 以上なら、a は 300 だという条件を書いてやれば良いんだ。

ball-3 で保存しよう。

## 6-ステップ 4：図形の動きを制御しようの続き

左からでた白丸を右端ではね返るように、手直ししてみよう。

```
1 int a=0;
2 int b=1;
3
4 void setup() {
5     size(600, 200);
6 }
7
8 void draw() {
9     background(150);
10    a=a+b;
11    if(a>=600) { [ ] ;}
12    circle(a, 100, 100);
13 }
```

### ヒント

右行きで、変数 a を使ったので、左行きは、別の変数 b にするんだ。もし a が 600 以上になつたら、a を減らすように変数を (b=-1;) にしてやれば、a=a-1 となって白丸は左に動くんじゃないかな。

### ミッション

- ① 右端では白丸が半分中に入っているね。表面ではね返らせよう。

ball-4 で保存しよう。

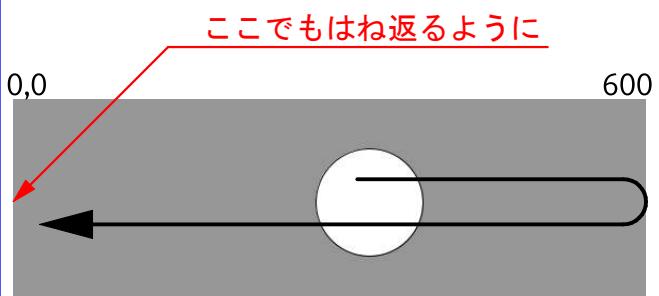
### 発展学習

左からでた白丸を両端ではね返るように、手直ししてみよう。

```
1 int a=0;
2 int b=1;
3
4 void setup() {
5     size(600, 200);
6 }
7
8 void draw() {
9     background(150);
10    a=a+b;
11    if(a>=550) { b=-1; }
12    if(a<=50) { [ ] ;}
13    circle(a, 100, 100);
14 }
```

### ヒント

左端に戻ってから再度右に進むには、a を増やすように変数を (b=1;) にしてやる条件を、もう 1 行加えればOKだ。



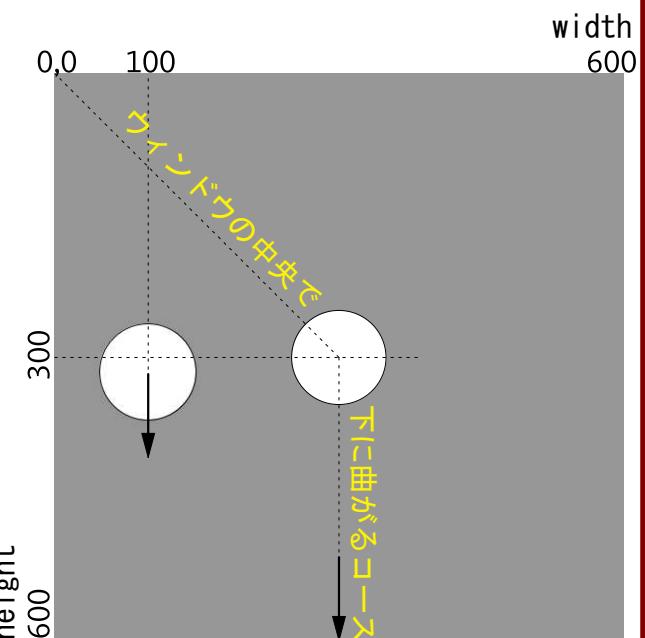
ball-5 で保存しよう。

## 6-ステップ 5：図形を自由に動かそう

**ball** を呼び出して、下のコードに手直しして、白丸を上から下に動かしてみよう。

```
1 int a=0;                                ball
2
3 void setup() {
4     size(600, 200);
5 }
6
7 void draw() {
8     background(150);
9     a=a+1;
10    circle(a, 100, 100);
11 }
```

```
1 int a=0;
2
3 void setup() {
4     size(600, 600);                      ← size を変えた
5 }
6
7 void draw() {
8     background(150);
9     a=a+1;
10    circle(100, , 100);
11 }
```



ball-6 で保存しよう。

### ミッション

- ① 白丸を左上(0, 0)から、右下(600, 600)へと動かそう。
- ② 白丸をウィンドウの中央で止めてみよう。

① のコードを手直しして、上図のように白丸をウィンドウの中央で下に曲げよう。  
X軸方向に変数 a を、Y軸方向に変数 b を使って、片方を中央で止めればOKだ。

```
1 int a, b=0;                            ← 変数に b を加えた
2
3 void setup() {
4     size(600, 600);
5 }
6
7 void draw() {
8     background(150);
9     a=a+1; b=b+1;
10    if (a>=) {a=;}
11    circle(a, b, 100);
12 }
```

ball-7 で保存しよう。

## 6-ステップ 6：おまけ

新規を開いて下のコードを打ってみよう。

```
1 int a;
2 int b=800;
3 void setup() {
4   size(800, 300);
5 }
6
7 void draw() {
8   background(255);
9   a=a+1;
10  b=b-1;
11  strokeWeight(5);
12  | line(a, 100, a+100, 100);      //右行きの矢
13  | line(a+40, 80, a+100, 100);
14  | line(a+40, 120, a+100, 100);
15
16  | line(b, 200, b-100, 200);      //左行きの矢
17  | line(b-40, 180, b-100, 200);
18  | line(b-40, 220, b-100, 200);
19 }
```

### ミッショントピック

右行きの矢が右端に消えたら、また左端から表れるようにしてみよう。  
左行きの矢が左端に消えたら、また右端から表れるようにしてみよう。

arrow で保存しよう。

### 次回の予告

プロセッシングが使えるようになった  
ので、次回は中間の復習をかねて、ア  
ニメーションを作って見よう。  
予習用紙を配るから、次回まで考て  
きてね。

See you next time!  
Have a beautiful day.

