

Processing

第 5 回



松田小学校 / 寄小学校

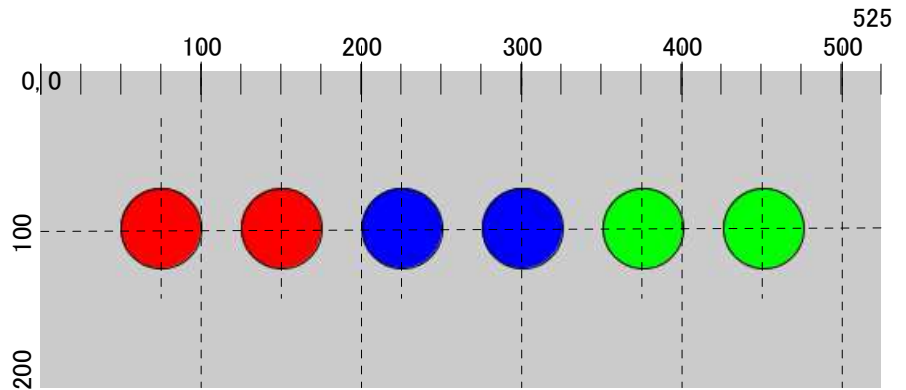
5 ステップ 0 : 前回の復習だよ

6balls を呼び出して実行してみよう。

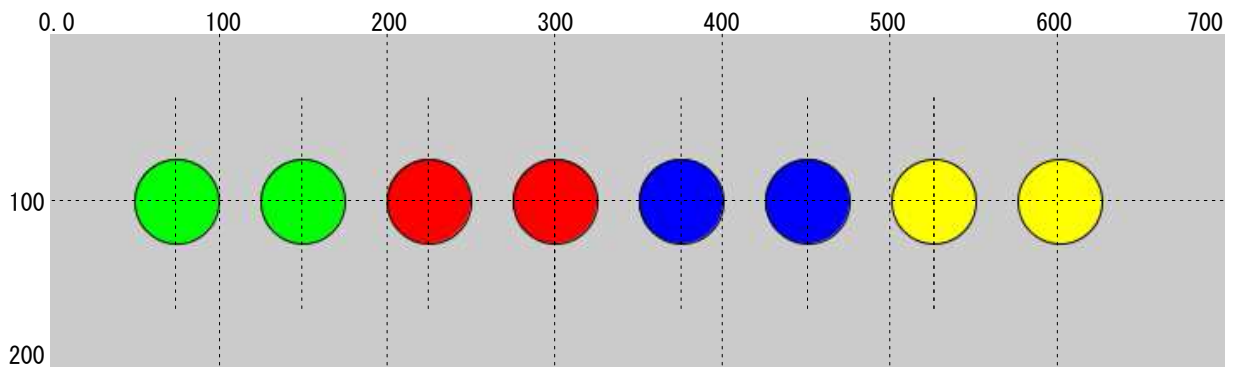
```

1 size(525, 200);
2
3 for(int a=1; a<=6; a=a+1) {
4   if(a<=2) {fill(255, 0, 0);}
5   if(a>=3) {fill(0, 0, 255);}
6   if(a>=5) {fill(0, 255, 0);}
7   ellipse(75*a, 100, 50, 50);
8 }

```



下の図のようにコードを変更してみよう。



空欄を埋めよう。

★重要だよ★

$a \% 2 == 0$ のとき、 a は だ。 ← 2 で割り切れる

$a \% 2 != 0$ のとき、 a は だ。 ← 2 で割り切れない

$\%$ は余り(あまり)を求める記号で、 $5\%2$ の答えは 1 だ。では、①~③はいくつだ？

① $3\%2 = \square$

② $10\%3 = \square$

③ $5\%7 = \square$

$$\begin{array}{r} 1 \dots \square \\ 2 \overline{) 3} \\ \underline{2} \\ 1 \end{array}$$

$$\begin{array}{r} 3 \dots \square \\ 3 \overline{) 10} \\ \underline{9} \\ 1 \end{array}$$

$$\begin{array}{r} 0 \dots \square \\ 7 \overline{) 5} \\ \underline{0} \\ 5 \end{array}$$

5-ステップ 1 : マウスを使ってみる

ファイルから新規を開き、次のコードを打って実行してみよう。

```

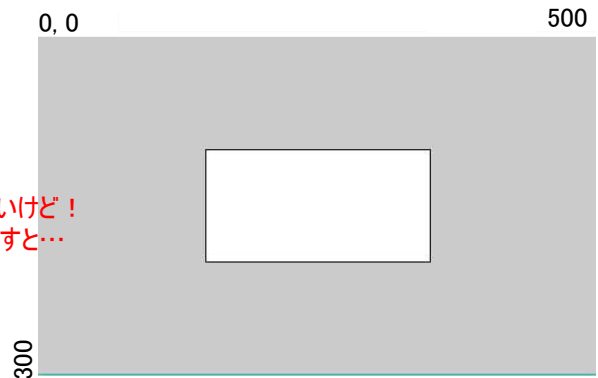
1 void setup() {
2   size(500, 300);
3 }
4
5 void draw() {
6   if(mousePressed)
7     fill(255, 0, 0);
8   rect(150, 100, 200, 100);
9 }
10
11

```

white-1 で保存しよう。

大文字だよ！

最初は白いけど！
マウスを押すと...



mousePressed はマウスが押されたという命令で、if () と結合して、もしマウスが押されたら、その行以下を実行せよとなるんだ。ウィンドウの上でマウスを押してみよう。

white-1 を下のコードに直して実行してみよう。

```

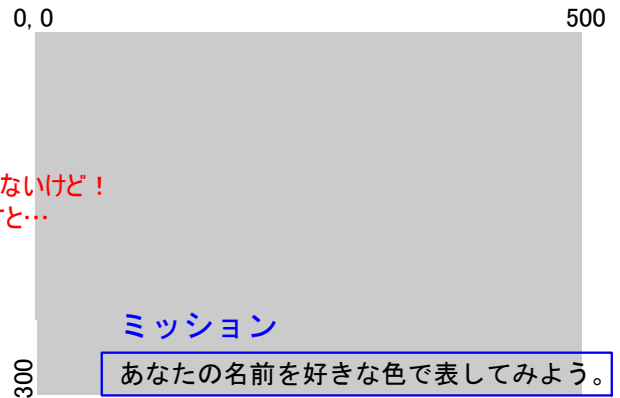
1 void setup() {
2   size(350, 250);
3 }
4
5 void draw() {
6   if(mousePressed)
7     {textSize(50);
8     fill(255, 0, 0);
9     text("Hello!", 100, 150);
10  }
11 }
12

```

white-2 で保存しよう。

大文字だよ！

最初は何もないけど！
マウスを押すと...



右のようなウィンドウが現れたはずだよ。
ウィンドウの上でマウスを押してみよう。

★超重要★ 動かすには、void setup (); と void draw (); が、

絶対に必要なんだ。void setup (); に続く { } の中の命令を一回だけ実行する。そして、void draw (); に続く { } の中の命令を何回も繰り返すから、動いているように見えるんだ。

```

1 int a=1;
2
3 void setup() {
4   size(200, 100);
5   frameRate(1);
6 }
7
8 void draw() {
9   background(0);
10
11   textSize(30);
12   text(a, 85, 55);
13   a=a+1;
14   if(a>=20) a=20;
15 }

```

ファイルから新規を開き、
左のコードを打って実行してみよう。

大文字だよ！

frameRate (?); とは、1秒間に何回表示するかを決める命令だ。() の数字が回数を決めており frameRate (1); は1秒に1回表示するんだ。

白と黒、それにグレーは省略形が使えるんだっかね。

a を 1 ずつ増やしていく命令だよ。

もし、a が 20 以上になったら、同じ文字を
繰り返し表示し止まっているように見せる命令だ。

a=a+1; の役割
a を次々に代入すると、

最初は int a=1; だ。
 2回目 a=1+1=2
 3回目 a=2+1=3
 4回目 a=3+1=4
 5回目 a=4+1=5
 6回目 a=5+1=6
 ...
 18回目 a=17+1=18
 19回目 a=18+1=19
 20回目 a=19+1=20

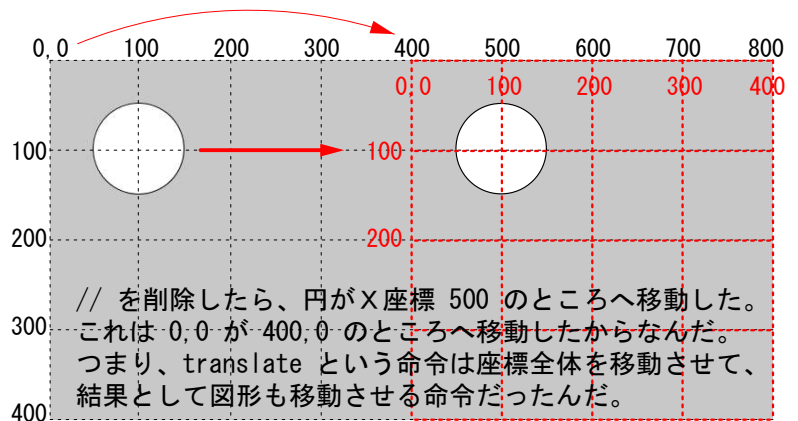
さあ、実行してみよう。正しくコードが書けていれば、驚くことがおきるよ！

5-ステップ 2 : 座標 (=図形) を移動させる

ファイルから新規を開き、左下のコードを打って実行してみよう。

```
1 int a;
2 void setup() {
3   size(800, 400);
4 }
5
6 void draw() {
7   background(200);
8   //translate(400, 0);
9   ellipse(100, 100, 100, 100);
10 }
```

下のようなウィンドウができたかな？
ふつうの円を、X座標 100、Y座標 100で描いただけだね。
じゃ、8行目の // を削除してから実行してみて。



translate は (400, 0) へと一気に移動させたけど、変数を使えば、ユックリ移動させることもできるんだ。
8行目を translate(a, 0); に書き換えて、10行目に a=a+1; を書き加えて実行してみよう。

ミッション-1

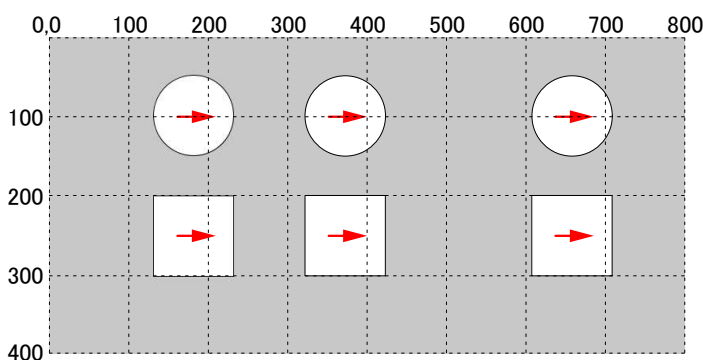
- ① a=a+1; を a=a+10; に書き換えて実行してみよう。a=a+1; はどんな働きをしていたのかな？
- ② int a=400; にして実行してみよう。どんな違いが表れたかな？
- ③ translate の X座標に変数を入れたから、X座標に平行に動いていった。ではY座標と平行に移動させるにはどうしたら良いのだろうか。size(800, 400); を size(800, 800); に変えて実行してみよう。
- ④ X座標とY座標の両方を変数にして実行したら、どうなるかな？ 実行する前に予想してね。

ミッション-1の解説

- ① a=a+1; は動きの速さをコントロールしていたんだね。加える数字が大きくなれば早くなるんだ。それともう1つ、a=a+1; は +1 の方向つまり + の方向へ移動させる命令だった。
- ② int a=400; は最初に図形が表示される座標値を表していたんだ。int a; は int a=0; の省略形だから、int a; のときはX座標が 0 から出発し、int a=400; のときはウィンドウの中央から表示されたんだ。
①を確認してみよう。int a=400; で a=a-1; とすると、どうなるかな？
- ③ translate の Y座標に変数を入れれば、Y座標と平行に下の方へ移動していくね。a=a-1; としたときと a=a+1; としたときでは、どんな違いがある？
- ④ X座標とY座標の両方を変数にして実行すると、斜め 45° の方向に移動を始めたね。これは X座標方向とY座標方向の速さが同じだから、中間の 45° の方向に行くんだよ。

ミッション-2

円の下に四角形を書き加えて動かしてみよう。



ミッション-2の解説

図形はいくつでも描けるけど、translate は座標全体を移動する命令だから、すべての図形を同じように移動させてしまうんだ。

1つ1つの図形を別々に動かすこともできるんだけど、translate では複雑なコードを書かないと無理なんだ。

個々の図形を動かす命令は、ステップ4で学ぶから楽しみにしててね。

trans で保存しよう

5-ステップ 3 : 図形を移動・回転させる

ファイルから新規を開き、左下のコードを打って実行してみよう。

```

1 float a;
2
3 void setup(){
4   size(500,500);
5 }
6
7 void draw(){
8   background(255);
9   a = a+.008;
10  //translate(250,250);
11  rotate(a);
12  line(0,0,150,0);
13  fill(255,0,0);
14  ellipse(150,0,10,10);
15 }
16

```

```

1 float a;
2
3 void setup(){
4   size(500,500);
5 }
6
7 void draw(){
8   background(255);
9   a = a+.008;
10  translate(250,250);
11  rotate(a);
12  line(0,0,150,0);
13  fill(255,0,0);
14  ellipse(150,0,10,10);
15 }
16

```

変数に小数が現れたぞ

白と黒、グレーは省略形が使えるんだね。

.008とは0.008の省略形だ

translateは移動せよという命令
(;)内には移動先の座標を書くんだ。

rotateは回転せよという命令
(;)内には回転の速さを指定するんだ。

ミッション

10行目の translate(250,250); が // によって効力を停止されているので、0,0 を中心に回転しているけど、// を消して移動させたらどうなるかな？

10行目の translate(250,250); は、0,0 の位置を250,250 に移動せよという命令だった。// を消して移動の効力を発生させたので、中心は250,250 へ移動し250,250 を中心に回転するようになったのだ。

ミッション

コードを変えて、いろいろな図形をさまざまに移動し回転させてみよう。
0,0 が250,250 に移動しても、図形への命令は回転の中心が0,0 としてコードを書いてね。

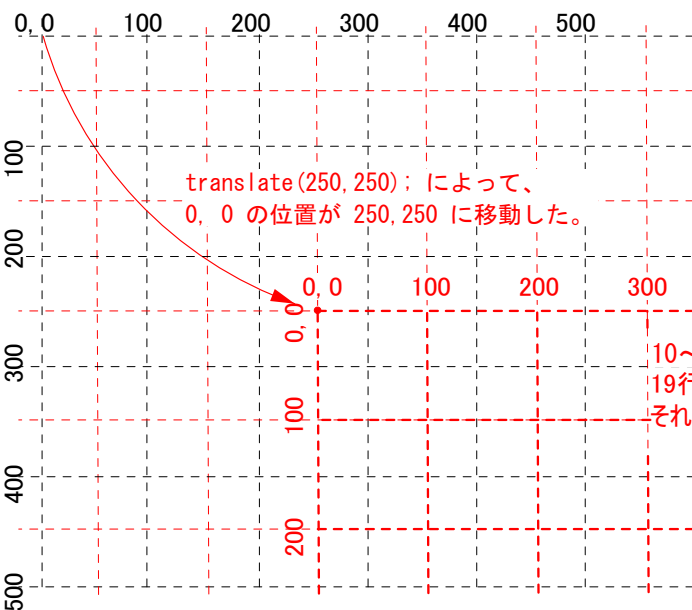
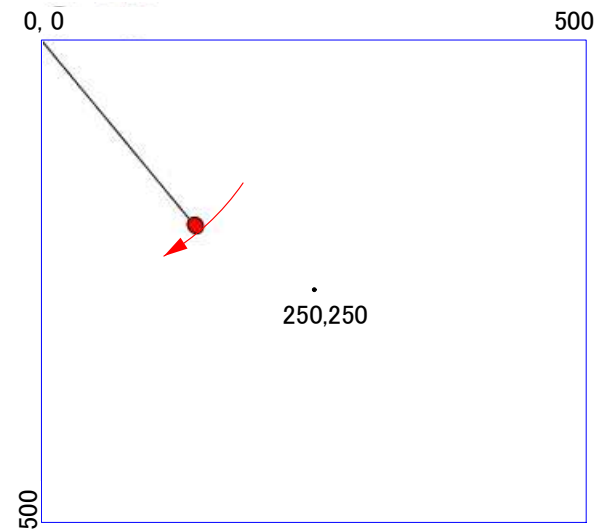
下のコードではどんな図形が動いているかな？

新しく打たないで、
上の watch_2 を修正だよ。

```

1 float a;
2
3 void setup() {
4   size(600,600);
5 }
6
7 void draw() {
8   background(255);
9   a = a+0.01;
10  translate(300,300);
11  rotate(a);
12  line(0,0,150,0);
13  fill(255,0,0);
14  ellipse(150,0,10,10);
15  line(0,0,100,150);
16  fill(0,0,255);
17  rect(100,150,20,20);
18
19  translate(50,50);
20  rotate(a);
21  line(0,0,150,0);
22  fill(0,255,0);
23  ellipse(150,0,80,80);
24  line(0,0,100,150);
25  fill(0,255,255);
26  rect(100,150,50,50);
27 }

```

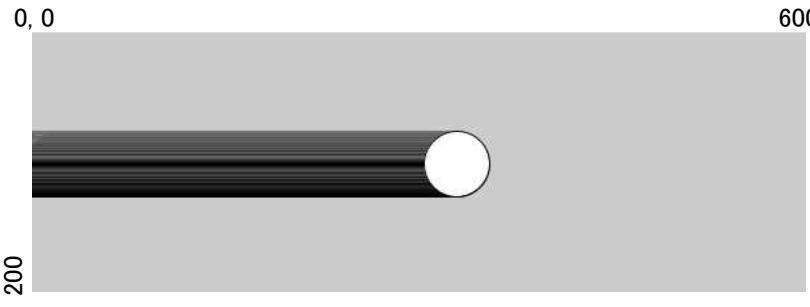


10~17行目をコピーして
19行目に貼り付け。
それから修正だ。

5-ステップ 4 : ひとつの図形を動かそう

ファイルから新規を開き、左下のコードを打って実行してみよう。

```
1 int a=0;
2
3 void setup(){
4   size(600,200);
5 }
6
7 void draw(){
8   a=a+1;
9   ellipse(a,100,50,50);
10 }
11
```



白い丸は動いているけど…

白い丸は右へと動いているけど、その後に黒い帯が付いている。これがあると、動いている感じがしないね。なぜ、黒い帯が出来てしまうか？ void draw の { } 内に書いた命令は何度も繰り返すといたけど1秒間に60回実行=表示しているんだ。X座標(▲)が a だから、a=a+1 で1ずつ右へ移動しながら、1秒間に60回実行=表示しているので、実行したすべての丸が残って帯のようになってしまうんだ。これを防ぐためには、背景も白い丸と同じ回数だけ書き換えてやれば、黒い帯は背景で隠れて一番手前の丸しか見えなくなるね。ではやってみよう。

```
1 int a=0;
2
3 void setup(){
4   size(600,200);
5 }
6
7 void draw(){
8   background(100);
9   a=a+1;
10  ellipse(a,100,50,50);
11 }
12
```

ball で保存しよう。

ミッション-1

background(100); を加えたので、最新の位置の白丸と背景が交互に表示されるようになった。そのため、1つの白丸が動いて見えるようになったのだ。それでは、白い丸を右から左に動かしてみよう。

- ヒント-1 a=a+1; は左辺の a に右辺の a+1 が代入され続けるから、a は1ずつ増え続けて、X座標が右へと移動していくんだ。とすれば、a=a+1; の式のなかに秘密がありそうだね。
- ヒント-2 右から左に動かすためには右から登場させないとダメだ。では、どうしたら良いかな？

ミッション-2 ファイルから新規を開いて

1. size(200,800); にして、X座標の中央を、半径 50 の円を縦に動かそう。
2. size(800,800); にして、半径 50 の円を 0.0 から斜め右下 45° に動かしてみよう。
3. size(800,800); のままで、50*50 の正方形を X・Y座標(600,600) から斜め左上 45° に動かしてみよう。

ミッション-2 は保存不要だよ。

丸の動きを制御しよう

a=a+1; が右へ動かしていたんだけど、もう1つ役割があった。それはスピードの制御だったね。a+1 の数字が大きくなるほどスピードが速くなるので、数字を変えて試して欲しい。

丸の動く距離を制御するには、if 文を使うんだ。まず、消えた丸を再び登場させよう。

```
1 int a=0;
2
3 void setup(){
4   size(600,200);
5 }
6
7 void draw(){
8   background(100);
9   a=a+1;
10  if(a>=600) a=0;
11  ellipse(a,100,50,50);
12 }
13
```

ミッション

a が 600 以上つまり右端より右になったら、a は 0 に戻れと指示したので最初に戻って左から再び白丸が登場したわけだ。

ではミッションです。丸が中央に来たときに、新しい丸が左から登場するようにするにはどうしたら良いだろう？

できたら、ball-1 という名前で保存しよう。

5-ステップ 5 : 図形の動きを止めてみよう

ファイルから ball-1 を開いてみよう。

```
1 int a=0;
2
3 void setup(){
4   size(600,200);
5 }
6
7 void draw(){
8   background(100);
9   a=a+1;
10  if(a>=width/2) a=0;
11  ellipse(a, 100, 50, 50);
12 }
```

ball-1 は中央に来た時に、新たな丸が左から登場するんだから、 $a > 600$ を $\text{width}/2$ 以上の時に、という条件に変えてやればいいんだ。(width はウィンドウの幅という意味だったね。)もちろん、 $a \geq 300$ でも正解だよ。

では、右端に来たら跳ね返って左へと戻るようにするには、どうしたら良いんだろうか？

ヒント

$a = a + 1$ が右へと動く命令だったね。そして、左へと動く命令は $a = a - 1$ だった。前は、別のファイルで両者は無関係に動かしたから、これで良かった。しかし、1つのファイルの中で、右へ行く命令と、左へ行く命令を同じ a で表すことはできない。そこで、もう1種類の変数 b を使うことにする。int $b = 1$ と宣言して、右端 width についた時に、 $b = -1$ とすれば $a = a + b$ に -1 が代入されて、 $a = a - 1$ となって、白丸は戻っていくのではないだろうか。できたら、ball-2 で保存しよう。

```
1 int a=0;
2 int b=1;           ball-2 だよ
3
4 void setup() {
5   size(600, 200);
6 }
7
8 void draw() {
9   background(100);
10  a=a+b;
11  if(a>=width) b=-1;
12  ellipse(a, 100, 50, 50);
13 }
```

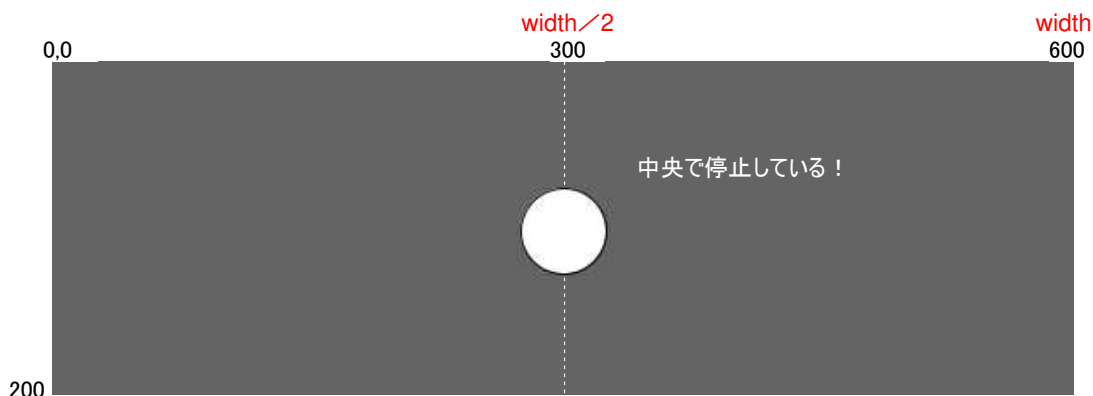
左のコードが ball-2 だ。右端に着くと、 $a = a + b$ に -1 が代入されて、 $a = a - 1$ となって、円は左へと戻っていくね。

では、中央で円が止まるようにするには、どうしたら良いのだろうか？ ステップ1で数字が大きくなるコードを書いたときに、20で止めるときに使ったやり方と同じだよ。中央に来たら、つまり a が $\text{width}/2$ 以上になったら、 a に 300 を代入してしまえば良いんだ。

右のコードが、中央で円が止まるコードだよ。中央までは $a = a + 1$ にしたがって、円は右へと進んでくる。しかし、 a が中央以上になると、 $a = 300$ が有効になるんだ。つまり、 $\text{ellipse}(300, 100, 50, 50)$; になって円は動かなくなる。コンピューターは相変わらず1秒間に60回の表示を繰り返しているけど、同じ場所で表示しているから止まっているように見えるんだ。

```
1 int a = 0;
2
3 void setup() {
4   size(600, 200);
5 }
6
7 void draw() {
8   background(100);
9   a = a + 1;
10  if(a >= width/2) a = 300;
11  ellipse(a, 100, 50, 50);
12 }
```

ball-stop で保存しよう。

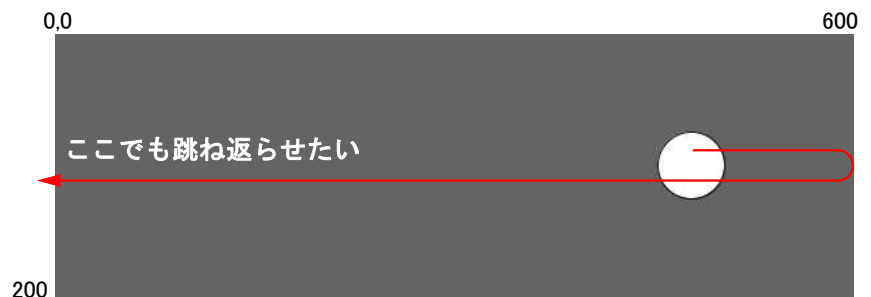


5-ステップ 6 : 図形を自由に動かそう

ファイルから ball-2 を開いてみよう。

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

ball-2 は往復したけれど、戻ってきたら終わっちゃったね。戻ってきても、また跳ね返って右に行くようにはできないだろうか？ 11行目と12行目の間に `if(a<=0)` ; を書き加えればOKだよ。



下が永久に往復するコードだ。

ミッション

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

- ① 左のコードは、円が壁の中まで入ってから、跳ね返っているね。壁の表面で跳ね返らなければおかしいから、壁の表面で跳ね返るように改良しよう。円の基準は中心だから半径分だけ手前で折り返せば良いね。
- ② 行きは白丸で進み、右端で折り返す時に赤丸に変わり、左端で折り返す時には青丸に変わるようにコードを変えてみよう。if(条件)に続くコードが複数のときは、そのコードを{ }でくくるんだ。
- ③ 行きが白色じゃなくて青くなってしまったね。変数 a の初期値が 0 だから、12行目の条件で青くなっているんだ。int a の初期値を半径の半分にしてみよう。

ball-3 で保存しよう。

ファイルから新規を開いて、下のコードを打って実行してみよう。

```
1 float a=1;
2
3 void setup(){
4   size(700, 700);
5 }
6
7 void draw(){
8   background(a, 0, 0);
9   a=a+0.5;
10
11
12
13 }
```

上のコードを実行すると

700×700のウィンドウが、だんだんと明るくなり、最後は真っ赤になるでしょう。a の初期値は 1 だから、R, G, B で(1, 0, 0)といえば、

もうほとんど黒色だよ。それが、a は 0.5 ずつ増えていき、最後は(255, 0, 0)で赤色になる。これも原理は、動く＝移動と同じだよ。

書き加えたのは、1. 名前の追加、2. 名前の大きさを変える、3. fullScreen で全画面にする。

左のコードに下記を加えてみよう。

1. 2行目→float b=1;
10行目→textSize(100);
11行目→text("あなたの名前", 50, 500); **半角のローマ字でね**
ここで一度実行してみてね。どうなった？
2. 次は改良と追加だ。
10行目→textSize(100);をtextSize(b);へ書き換える。
12行目→b=b+0.5; を追加。
ここでまた実行して、確認してね。
3. 次も改良だね。
4行目→size(700, 700);をfullScreen();へ書き換える。
さあ、どうなったかな？

name で保存しよう

全画面に戻すには、どこかでマウスをクリックして左上のESCを押してね。

5-ステップ 7 : 図形をマウスで動かそう

ファイルから新規を開き、左下のコードを打って実行してみよう。
マウスで図形を動かすこともできるよ。

```
1 void setup(){
2   size(800,600);
3 }
4
5 void draw(){
6   rect(mouseX, 200, 50, 30);
7 }
```

大文字だよ

X座標はマウスで決めます、という指示。

ステップ3では自動で動くようにするために変数が必要だった。人間がマウスを操作して動かすなら、変数はいらぬよね。しかしやっぱり四角形がいくつもかけてしまい、古い四角形が消えないことは同じだ。そこで工夫しよう。

```
1 void setup(){
2   size(800,600);
3 }
4
5 void draw(){
6   background(100);
7   rect(mouseX, 200, 50, 30);
8 }
```

加えた

人間が操作するとたやすく、しかも自由に動かすことができるね。Y座標のほうも `mouseY` を入れてみよう。これでウィンドウ全体を自由に動かすことができるようになった。四角形に色選択を使って色付けしてみよう。

```
1 void setup(){
2   size(800,600);
3 }
4
5 void draw(){
6   background(100);
7   fill(0,255,0);
8   rect(mouseX,mouseY,50,30);
9 }
```

四角形には、X座標とY座標、それに横幅、縦の大きさの4つの数字が入った。しかし、第5の数字を入れることができる。その働きは何か？ 縦の大きさを50にしてから、実際に数字を入れて確かめてみよう。最初は1桁の数字を入れて、だんだん大きくしてみよう。

30を50にして、をいいて、1桁の第5の数字を入れる。

第5の数字が大きくなるに従って、だんだん丸みが大きくなってきたね。では、9行目に直径100の丸を描いて比べてみよう。ただし、`mouseX,mouseY` を使うと、四角形と重なってしまうから、`(mouseX-200,mouseY-200,100,100)` にして四角形と丸をはなしてやろう。

mouse で保存しよう。

次回の予告 乱数と色をやるよ。



Bye-bye!