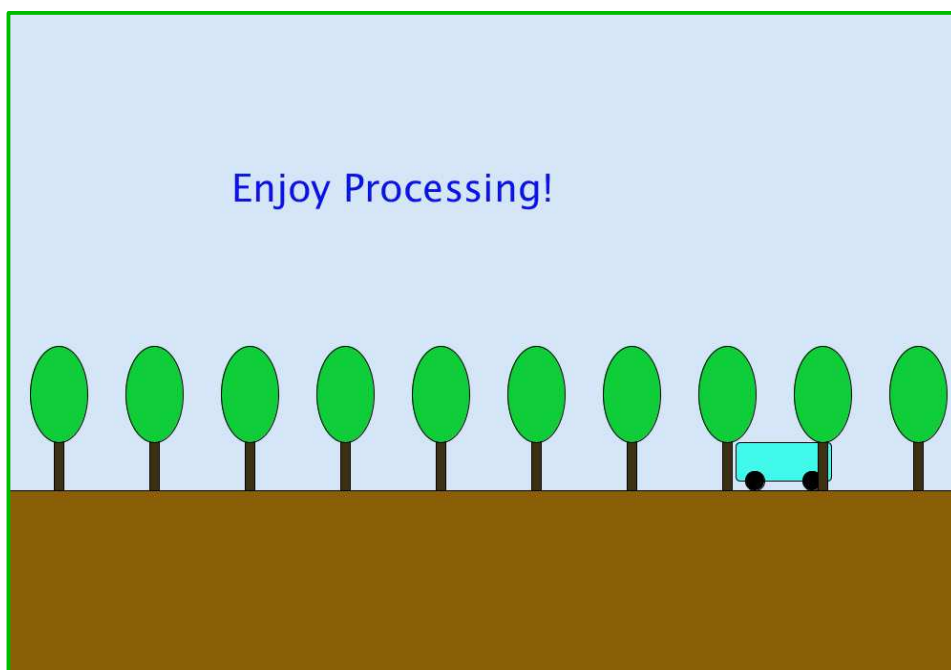


Processing

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。保存していなかったら、

ファイル→新規で、打ってみよう。

```

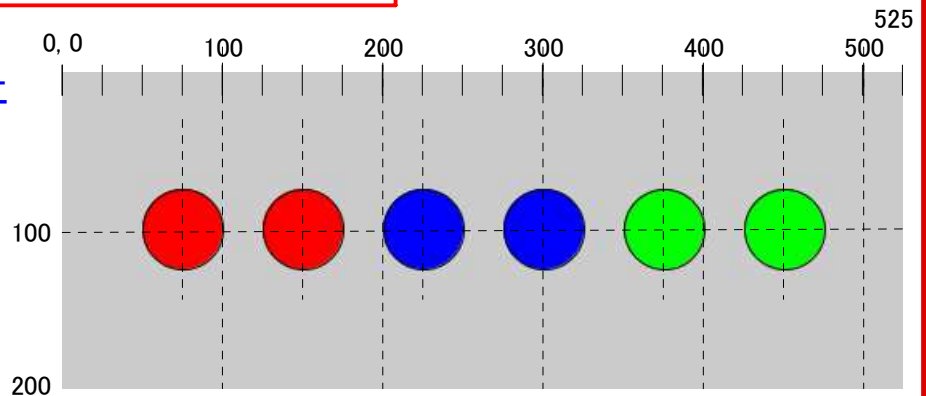
1 size(525, 200);
2
3 for(int a = 1; a <= 6; a=a+1) {
4     if(a<=2) {fill(255, 0, 0);}
5     if(a>=3) {fill(0, 0, 255);}
6     if(a>=5) {fill(0, 255, 0);}
7     ellipse(75*a, 100, 50, 50);
8 }
    
```

コード

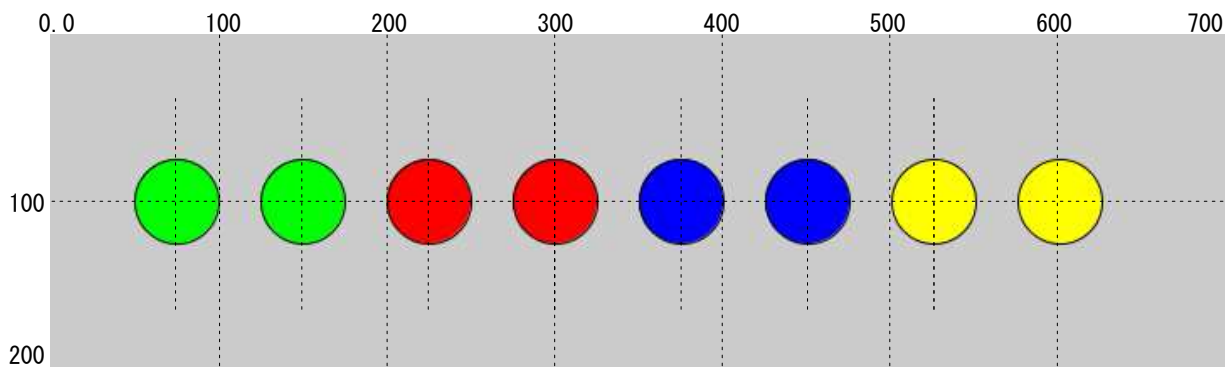
6balls の呼び出し方

ファイル→開く→USB
→6balls をクリック

右の図のようになっただかな？



下の図のように、黄色の丸を2つ足す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 $3\%2 = \text{□}$

② $10\%3 = \text{□}$

③ $5\%7 = \text{□}$

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

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

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

println(); を使ってみよう。

5 ステップ 1 : マウスを使ってみるー動かす準備

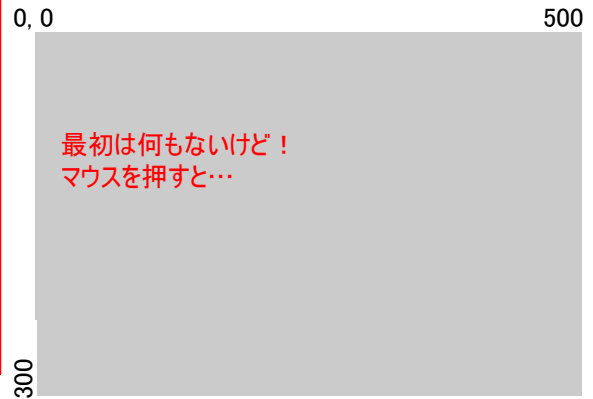
ファイルから新規を開き、次のコードを打って実行してみよう。

`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。

```
1 void setup() {          コード
2   size(500, 300);
3 }
4
5 void draw() {
6   if(mousePressed) {
7     textSize(50);
8     fill(255, 0, 0);
9     text("Hello", 100, 150);
10  }}

```

大文字だよ!



ミッション

Hello に代えて、あなたの名前を好きな色で表してみよう。

white で保存しよう。

★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。
`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;          コード
2
3 void setup() {
4   size(200, 100);
5   frameRate(3);
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 }

```

大文字だよ!

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。

`frameRate(?);` とは、1秒間に何回表示するかを決める命令だ。
`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 になったら、同じ文字を繰り返し表示して、止まっているように見せる命令だ。

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!

number で保存しよう。

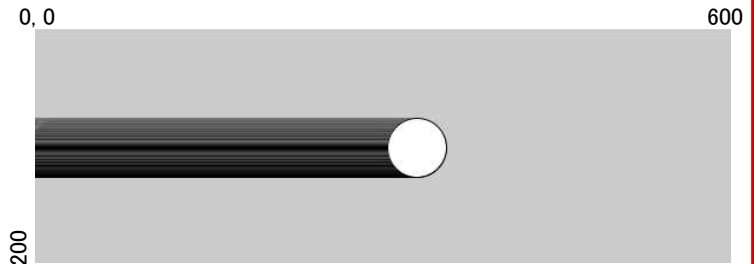
5 ステップ 2 : ひとつの円を動かそう

```

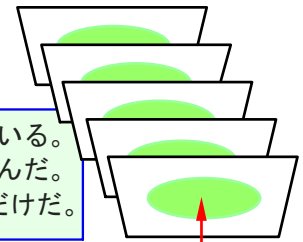
1 int a=0;
2
3 void setup() {
4   size(600, 200);
5 }
6
7 void draw() {
8
9   a=a+1;
10  ellipse(a, 100, 50, 50);
11 }

```

ファイルから新規を開き、コードを打って実行してみよう。



白い丸は動いているけど…黒い帯が！



void draw の { } 内に書いた命令は、すべて1秒間に60回実行=表示している。X座標()が a だから、a=a+1 で1ずつ右へ移動しても、前の丸が全部残るんだ。背景(background)を入れてやれば、前の丸は背景で隠れて新しい丸が見えるだけだ。

上のコードを手直ししてみよう。

```

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 }

```

ミッション-1

background(100); を加えたので、最新の白丸以外は背景に隠れて、最新の白丸が1つだけ見えるようになった。では、白丸を右から左に動かしてみよう。

ヒント-1 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ね。int a=0 は a が 0 で、0 は左端だから、右端はどう表すかな？

ヒント-2 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; の記号を変えると、左へ動かそうだね。

見る方向

ball で保存しよう。

上のコードを手直しして、白丸が左から表れて右に消えたら、

また左から表れるようにしてみよう。

```

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

```

ヒント

白丸を左から右に動かすには、白丸が左端から出発するんだから a は 0 だね。

また白丸を左から表すためには、もし a が 600 以上になったら、a は 0 だ、という条件を書いてやれば良いんだ。

条件を加えた

ball-1 で保存しよう。

5 ステップ 3 : 図形の動き制御してみよう

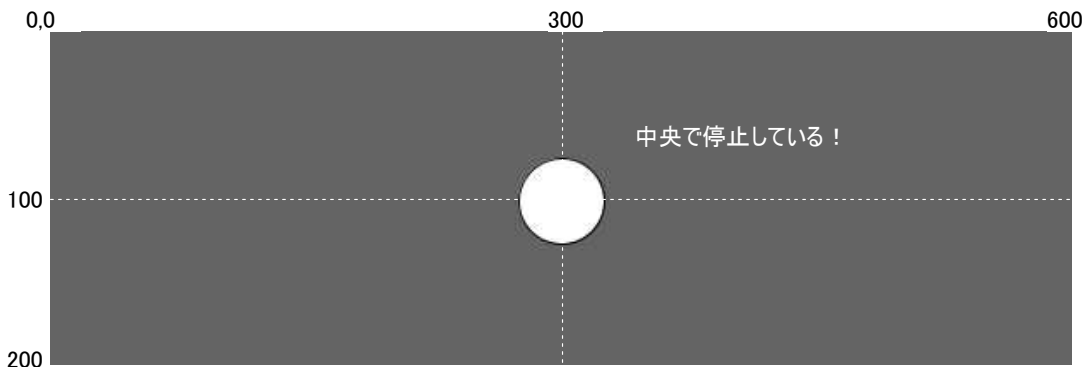
1 `int a=0;` `ball-1` を手直して、右に進んでいる白丸を中央で止めてみよう。

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

左のコードが、中央で白丸が止まるコードだよ。
中央までは `a=a+1` にしたがって、白丸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、`ellipse(300,100,50,50);` になって白丸は動かなくなる。
`X=300、Y=100` という同じ場所で、1秒間に60回も表示しているから、止まっているように見えるんだ。
ステップ1の `number` が 20 で止まったのと同じ原理だ。

条件文の中にコードを書いて、白丸を止めてみよう。

ball-stop で保存しよう。



`ball-stop` を手直して、右に進んでいる白丸を、右辺で跳ね返って左へ戻してみよう。

`a=a+1` が右へと動く命令で、左へと動く命令は `a=a-1` だった。前回までは別のファイルだっ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。もう1種類の変数 `b` を使うんだ。`int b=1` と宣言して、右端 `width` についての時に `b=-1` とすれば `a=a+b` の `b` に `-1` が代入されて、`a=a-1` となって、白丸は戻っていくだろう。

```
1 int a=0;  
2 int b=1;
```

変数 `b` を書き加えた。`b=0` だと、10行目の `a=a+b` は `a=0+0` になってしまうよね。
つまり、`a` が 0 だから白丸は左辺から動かないんだ。

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

左のコードが 往復するコードだ。
右端に着くと、`a=a+b` の `b` に `-1` が代入されて、`a=a-1` となって、白丸は左へと戻っていく。

変数 `a` に `b` を加えたコードだ。`a=0、b=1` だから、`a=0+1` になるね。

ball-2 で保存しよう。

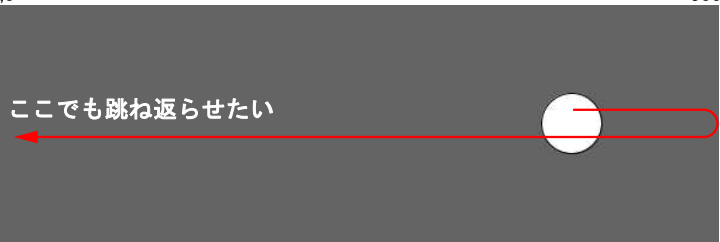
5 ステップ 4 : 図形を自由に動かそう

```

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
13  ellipse(a, 100, 50, 50);
14 }

```

ball-2 の続きだよ。左辺でも跳ね返らせたいたい。



0,0 width 600
200

ここでも跳ね返らせたいたい

ball-2 は往復したけれど、戻ってきただけだったね。戻ってきても、また跳ね返って右に行くようにしよう。a が 0 以下なら、b が 1 という条件を加えるんだ。11行目と12行目の間に `if(a<=0) {b=1;}` ; を書き加えれば OKだよ。

上の永久に往復するコードを手直しして、端でもぐらないようにしよう。(1~6行目までは同じ)

```

8 void draw() {
9   background(100);
10  a=a+b;
11  if(a>= ) {b=-1;}
12  if(a<= ) {b=1;}
13  ellipse(a, 100, 50, 50);
14 }

```

ヒント

上のコードは、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。

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どうなった？ 次は右だ。

左のコードに下記を加えてみよう。

- 2行目→`float b=1;`
10行目→`textSize(100);`
11行目→`text("あなたの名前", 50, 50);`
名前は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？
- 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。
- 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？

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。

name で保存しよう

5 ステップ 5 : 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 }
```

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。今度はあなたがマウスを操作して動かすから、変数はいらないよね。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。

大文字だよ

X座標はマウスで決めます、という指示。

上のコードを手直して、影を消してみよう。そして、Y座標方向にも動かそう。

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

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。

でも、黒い影ができてしまうね。6行目にコードを加えて、影を消してみよう。

上のコードを手直して、四角に色を付けてみよう。そして、ミッションをやってみよう。

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

ミッション

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。

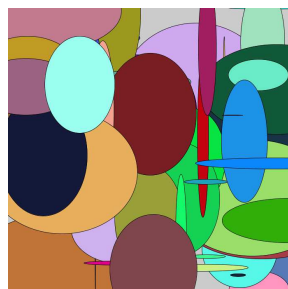
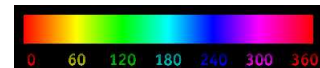
最初は1桁の数字を入れて、だんだん大きくしてみよう。

30 の後ろに , をおいて、1桁の第5の数字を入れる。

mouse で保存しよう。

次回の予告

次回は乱数 (=random) と色を扱うよ。



Bye-bye!