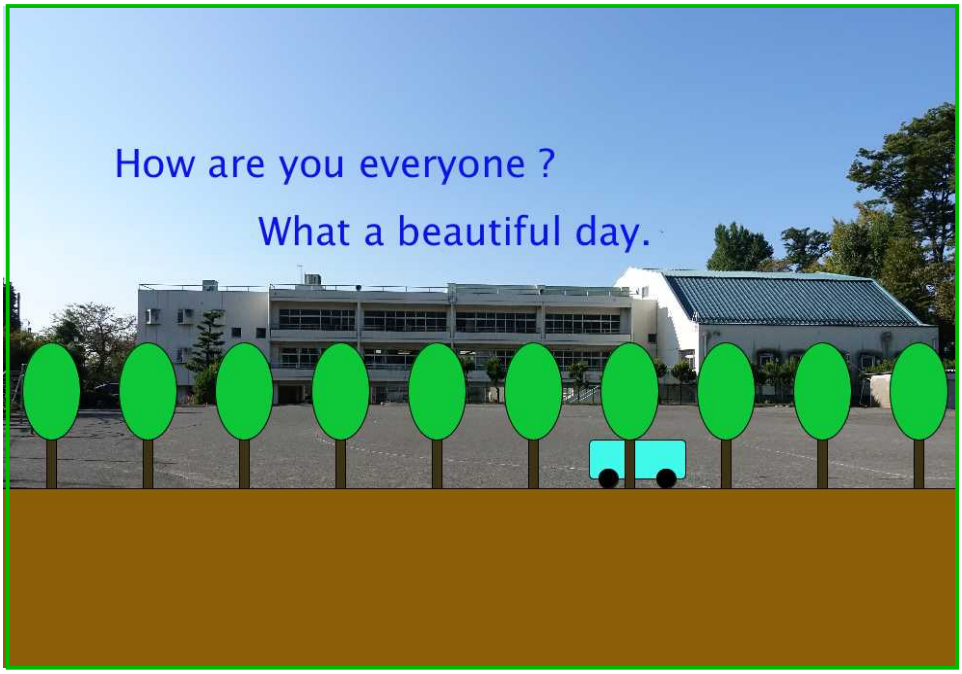
Processing

第６回



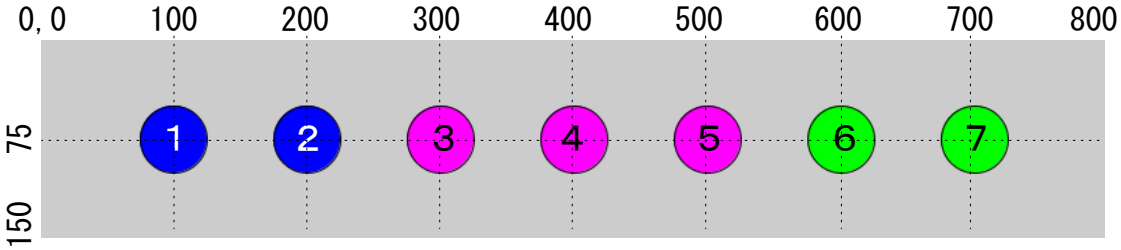
松田小学校／寄小学校

６ーステップ　０：前回の復習だよ

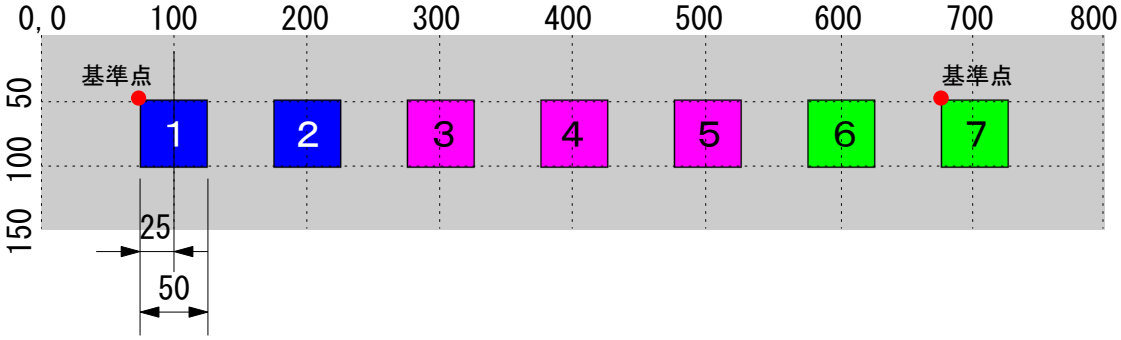
ファイル→新規で打ってみよう。

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

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。

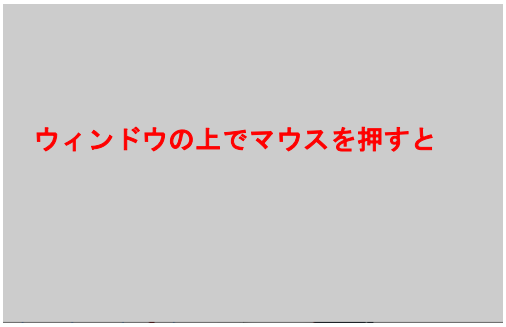


rect-7 で保存しよう。

６ーステップ　１：マウスを使ってみようー動かす準備

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   }}
```



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。

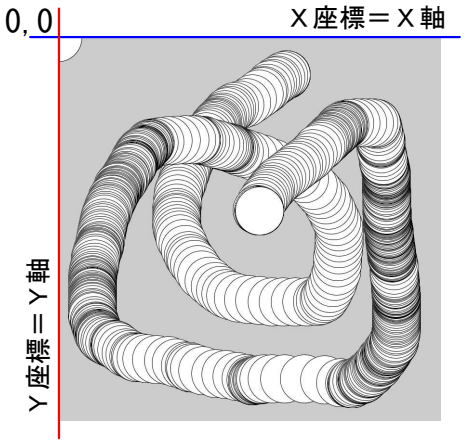
```
1 void setup() {
2   size(800, 800);
3 }
4
5 void draw() {
6   circle(mouseX, mouseY, 100);
7 }
```

{ }内を最初に１度だけ実行する

{ }内を何度も実行をくり返す

void setup＝準備
（ヴォイド　セットアップ）

void draw＝描く
（ヴォイド　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ーステップ　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。

```
int a=0;

void setup(){
  size(600, 200);
}

void draw(){
  a=a+1;
  circle(a, 100, 100);
}
```

0,1,2,3,4,5～と代入され、Ｘ座標が大きくなって右へ動いていく。

★超重要★

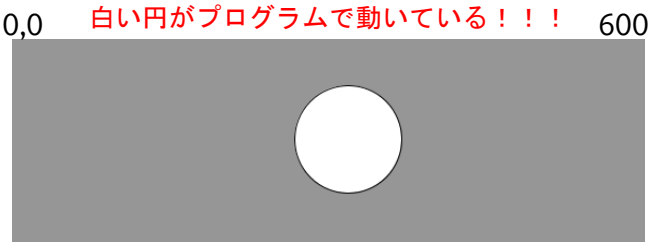
図形を動かすには、void setup(); と　void draw(); が、絶対に必要なんだ。

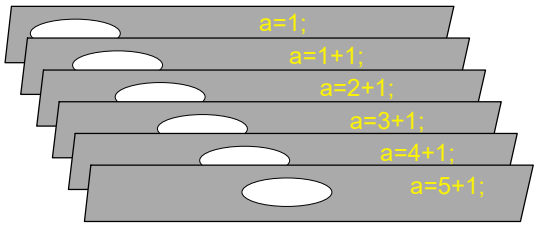
void setup() に続く　{　　}　の中の命令を１回だけ実行し、その後、
void draw() に続く　{　　}　の中の命令を何度も（１秒間に60回）実行するから、
動いているように見えるんだ。

黒い帯を取るよう手直ししよう。

```
int a=0;

void setup(){
  size(600, 200);
}

void draw(){
  background(150);
  a=a+1;
  circle(a, 100, 100);
}
```

加えた





グレーのバックグラウンドで隠れて
後ろの円が見えないので、円が動いて見えるね。

見る方向

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

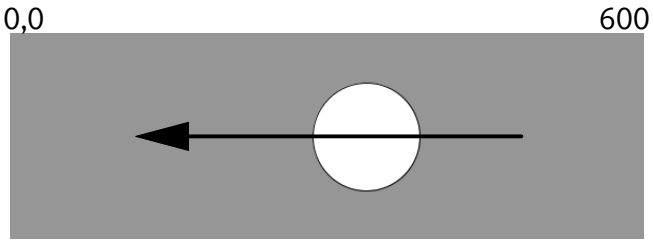
６ーステップ　３：図形の動きを制御しよう

白丸が左から表れるようにしてみよう。

```
int a=      ;

void setup() {
  size(600, 200);
}

void draw() {
  background(150);
  a=a   1;
  circle(a, 100, 100);
}
```

初期値（＝白丸が最初にある位置）

１ずつ小さくなる



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。

```
int a=0;

void setup() {
  size(600, 200);
}

void draw() {
  background(150);
  a=a+1;
  if(a>=      ) {a=0;}
  circle(a, 100, 100);
}
```

a を 0 に戻した

条件を加えた

ヒント

白丸をまた左から表すには、もし a が 600 以上になったら、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。

```
int a=0;

void setup() {
  size(600, 200);
}

void draw() {
  background(150);
  a=a+1;
  if(a>=      ) {a=      }
  circle(a, 100, 100);
}
```

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ーステップ 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。

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

void setup() {
  size(600, 200);
}

void draw() {
  background(150);
  a=a+b;
  if(a>=600) { [　　];}
  circle(a, 100, 100);
}
```

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 以上になったら、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。

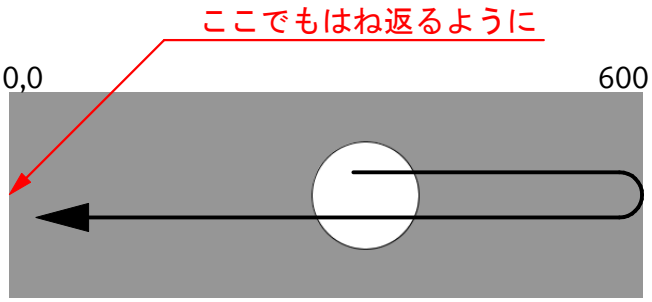
発展学習

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。

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

void setup() {
  size(600, 200);
}

void draw() {
  background(150);
  a=a+b;
  if(a>=550) { b=-1 ;}
  if(a<=50) { [　　];}
  circle(a, 100, 100);
}
```

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ーステップ ５：図形を自由に動かそう

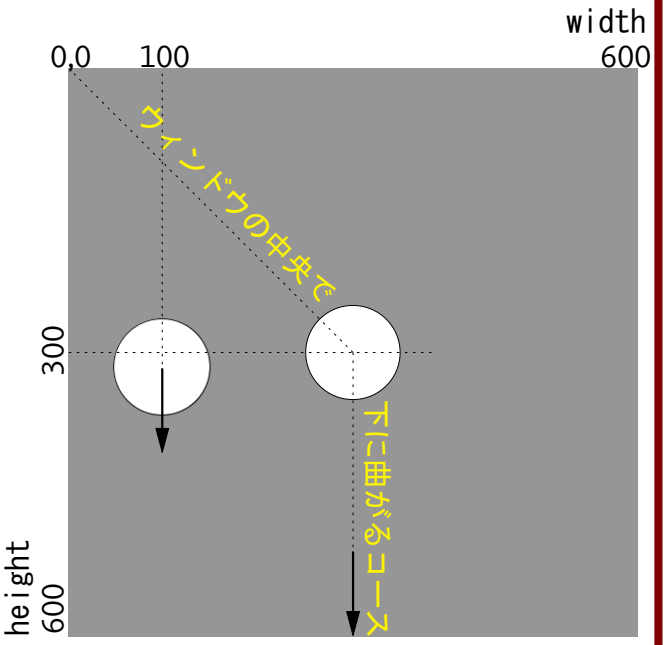
ball を呼び出して、下のコードに手直しして、白丸を上から下に動かしてみよう。

ball

```
int a=0;

void setup(){
  size(600,200);
}

void draw(){
  background(150);
  a=a+1;
  circle(a,100,100);
}
```

```
int a=0;

void setup(){
  size(600,600);
}

void draw(){
  background(150);
  a=a+1;
  circle(100,□,100);
}
```

size を変えた



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。

```
int a,b=0;

void setup(){
  size(600,600);
}

void draw(){
  background(150);
  a=a+1; b=b+1;
  if(a>=□){a=□;}
  circle(a,b,100);
}
```

変数に b を加えた

ball-7 で保存しよう。

６ーステップ ６：おまけ

新規を開いて下のコードを打ってみよう。驚くべきことがおきるよ。

```
int a=0;

void setup(){
  size(200,100);
  frameRate(3);
}

void draw(){
  background(0);
  a=a+1;
  if( a>=20 ){ a=20; }
  textSize(30);
  text(a,85,55);
}
```

frameRate(?); とは１秒間に何回表示させるかを決める命令だ。何も書かなければ、１秒間に６０回表示させる。

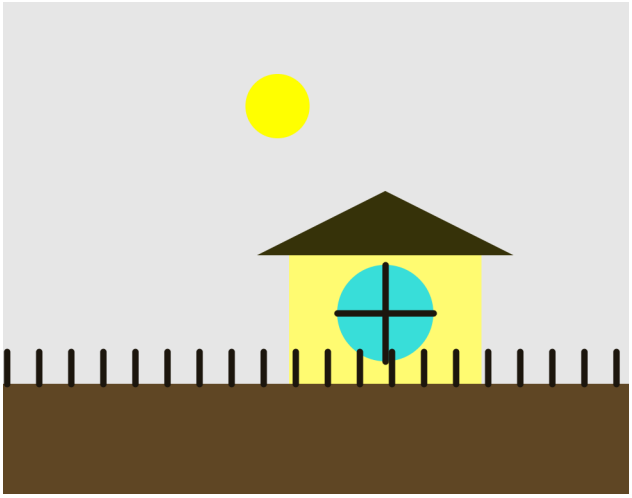
ミッション

① 数字が 1→２０ まで大きくなっていったね。今度は ２０→１ へと小さくしてみよう。どこを変えれば良いかな？ ヒントは４ヶ所だ。

number 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、次回まで考えてきてね。



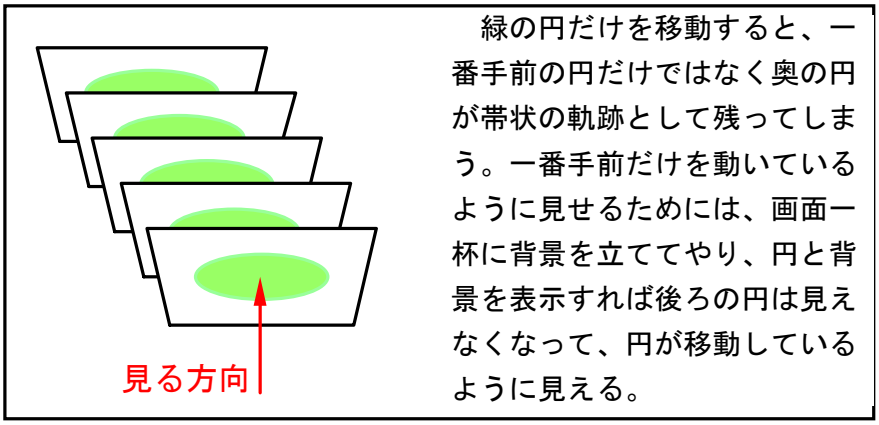
See you next time!
Have a beautiful day.

伴走の方へ　　第５回

① for 文では、ellipse と rect での基準点の違いに注意して下さい。第３回のステップ３でも扱っていますが、同じ位置に表示するためには、ellipse では変数を掛けるだけだったものが、rect では変数を掛けてから、四角形の辺長の半分を引く必要があります。X座標の 100*a － 25 です。

```
size(800, 150);
for(int a=1; a<=7; a=a+1){
  if(a<=2){fill(0, 0, 255);}
  if(a>=3){fill(255, 0, 255);}
  if(a>=6){fill(0, 255, 0);}
  rect(100*a-25, 50, 50, 50);
}
```

② フィルム映画が 1 秒間に２４コマ(コマを frame という)映写することによって動きを実現しているように、プロセッシングは１秒間に６０コマ表示しています。



フィルム映画と違ってコマが送られないため、すべてがウィンドウに重なって表示されています。それを無理に表示するとステップ１とかステップ２のように、いくつもの図形が描かれます。そこで、１枚の図形とその直後の図形のあいだに背景＝ background を挟んで、最新の図形と 1 つ後ろの図形を切断して、繰り返し表示される最新の図形だけを、パラパラ漫画のようにして動くように見せています。

③ ステップ１で記したように、動かすには void setup(); と void draw(); というコードが不可欠です。しかし、動きがなくても void setup(); と void draw(); を使って書いても大丈夫で、ふつうに表示されます。

④ コマンドに続く( ) には、いくつかの数字が入ります。これは座標値や色番号などといった、同じ性格をもった数字であることが多いのですが、rect の隅丸指定数字や色指定の透明度のように、中にはまったく性格の違う数字が入ることがあります。ヘルプからレファレンス(英語です)とたどって調べて下さい。日本語をという方は http://www.technotype.net/processing/reference/index_ext.html　をどうぞ。

⑤ 条件分岐の真偽値判断には、true や false が使われることが多いのですが、馴染みが悪く感じます。そのため、本ガイドでは、true を yes、false を no で表しています。もちろん yes の場合のみコードは実行されます。

⑥ void setup( ){ に書いたコードは１回だけ実行されますが、void draw( ) のなかでも有効に働いています。たとえば、noFill( ); とかけば、void draw( )の中を含めて以下の図形すべてに有効なので、無効にするためには再度 fill( ); を書いてやる必要があります。

⑦ 動かす原理は、動かそうとする図形の座標や大きさなどに変数を使い、その変数を a=a+1; などの式によって増減させて、座標値を変えることによって動かします。動きの早さは a=a+x; の x に足す(引く)数字によって調節します。小数を使いたいときは、変数宣言は float で行います。なお、後述の frameRate(); でも早さの調節ができます。

⑧ 変数には１つの役割しか与えることが出来ません。右行きで変数 a を使ったら、左行きには別の変数を使う必要があります。同様に、横直径と縦直径に同じ変数を使うと、真円のまま拡大・縮小します。横直径と縦直径に別の変数を使えば、楕円として拡大・縮小します。

⑨ 斜め４５°に動かすには、Ｘ座標とＹ座標ともに同じ変数を使えば、ＯＫです。しかし、４５°以外の角度で動かすには、Ｘ座標とＹ座標で違う変数を使う必要があります。また、ステップ５のように途中で止めたりするには、Ｘ座標とＹ座標で違う変数をつかって、片方だけ止めるように別々に制御してやります。

⑩ コマを１秒間に何回表示させるかの命令が frameRate(); で、カッコ内に何も書かなければ 1 秒間に６０コマ表示し、数字を書けばその数字分の回数を表示させます。小数以下の数字も記入できます。最初に１度だけ決めれば良いので、void setup(); の下に書きます。frameRate(); の（ ）内の数字で動きの速さを調節できます。

ステップ　１　ミッション①、②

```
void setup(){
  size(800,800);
}

void draw(){
  //background(100);
  ellipse(mouseX,300,100,100);
}
```

Ｙ座標には mouseY を

③

```
void setup(){
  size(800,800);
}

void draw(){
  //background(100);
  ellipse(400,400,mouseX,mouseY);
}
```

ステップ　４　ミッション①

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

void setup(){
  size(600,200);
}

void draw(){
  background(150);
  a=a+b;
  if( a>=600-50 ){b=-1;}
  ellipse(a,100,100,100);
}
```

ステップ　５　ミッション①

```
int a=0;

void setup(){
  size(600,600);
}

void draw(){
  background(150);
  a=a+1;
  ellipse(a,a,100,100);
}
```

ステップ　５　ミッション②

```
int a=0;

void setup(){
  size(600,600);
}

void draw(){
  background(150);
  a=a+1;
  if(a>=width/2){a=width/2;}
  ellipse(a,a,100,100);
}
```

ステップ　５　ミッション：真ん中で下方向へ

```
int a,b=0;

void setup(){
  size(600,600);
}

void draw(){
  background(150);
  a=a+1;    b=b+1;
  if(a>=width/2){a=width/2;}
  ellipse(a,b,100,100);
}
```

ステップ　６　ミッション①

```
int a=21;

void setup(){
  size(200,100);
  frameRate(3);
}

void draw(){
  background(0);
  a=a-1;
  if( a<=0 ){ a=0; }
  textSize(30);
  text(a,85,55);
}
```