Processing

第 9 回

松田小学校／寄小学校

９ーステップ　０：前回の復習だよ

　画像をファイルを開いて、p503.jpg の写真と牛のイラスト画像 bull.png をテキストエリアに引っ張ってこよう。そして、下のコードを打ってみよう。

```
1  PImage p503,bull;
2  float a=0;
3
4  void setup(){
5    size(      ,      );
6  }
7
8  void draw(){
9    p503 =loadImage("p503.jpg");
10   image(p503,0,0);
11
12   bull=loadImage("bull.png");
13
14   if(a>150)a=150;
15     image(bull,180,180,a,a);
16      a=a+0.5;
17 }
```

p503 と bull を使うという宣言だね。

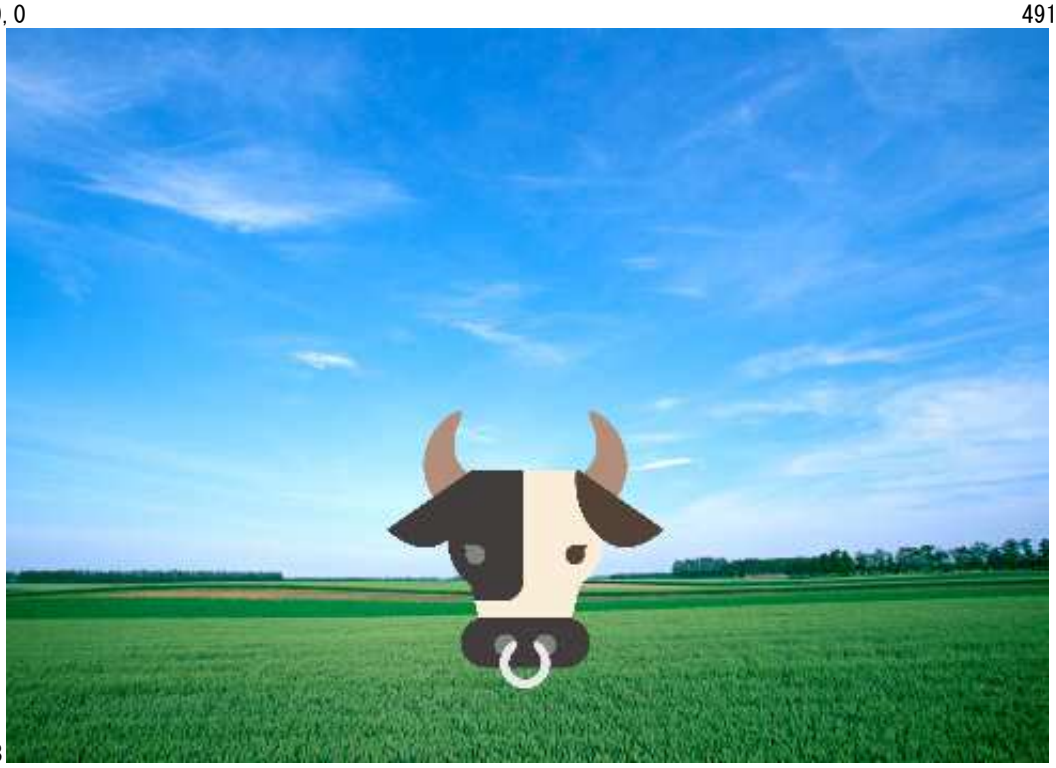
p503 の写真サイズ(491,348)に合わせたんだ。

p503 を背景に使っているよ。

牛さんを呼び出そう。

牛さんがウィンドウの外にはみ出さないところで止めよう。

変数 a を使って、牛さんの大きさをを変えるよ。



９ーステップ　１：if と動きを復習してみよう

if はとても便利な命令だよ。図形の動きを、if であやつろう。

新規を開いて下のコードを打って実行してみよう。　　ミッション−１

```
1  int a;
2  void setup() {
3    size(800, 800);
4  }
5
6  void draw() {
7    ellipse(a, 150, 100, 100);
8    a=a+1;
9  }
```

実行してみよう

黒い帯がでてしまった。これを消すには、background();を加えれば良かったね。どこに書けば良いのだろうか、考えて書き加えてみよう。書き加えたら①〜③を実行してみよう。
① ウィンドウの中央にきたら赤色にしてみよう。a=a+1; の下に、もし a が中央以上なら、赤色に塗れと書けば良いんだ。
② ウィンドウの中央で、直径200の円にしてみよう。赤色に塗れの後に、直径200の円を書く命令を書けば良いんだ。
③ ウィンドウの中央で、直径200の円を下向きに方向変更しよう。円のＹ座標に、変数 a を入れてやればＯＫだ。
でも下行きの時は、Ｘ座標は変数じゃないから直してね。

```
1  int a;
2  void setup() {
3    size(800, 800);
4  }
5
6  void draw() {
7    background(255);
8    ellipse(a, 150, 100, 100);
9    a=a+1;
10
11   if(a>=width/2) {fill(255, 0, 00);
12   //background(255);
13   ellipse(400, a-250, 200, 200);
14 }}
```

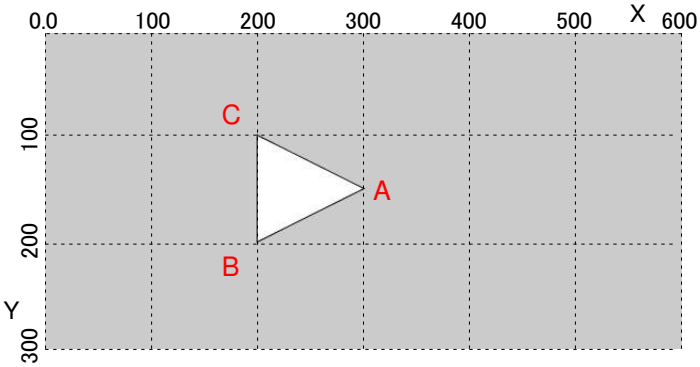
手直ししたのが左下のコードだ。
① のコードは、１１行目だ。
② のコードは、８行目コードを１３行目にコピペで挿入して、直径を変えればＯＫだね。
③ は、１３行目のコードが表しているよ。下向きだからＹ座標に変数 a を入れるんだね。でも、a-250 となっているのはなぜだろ？　　答えはこの回の最後にあるよ。

１３行目のコードを書くと赤色の円は下に行くけど、右に行く円も残ったままだね。これは下向きの円の命令は書いたけど、前からの右向きの円には何も指示してないから、以前の動きを続けているんだ。そこで１２行目の background を有効にしてやると、右へ行く円は消えるよ。

red-ball で保存しよう。

ミッション−２

新規を開いて下の三角形のコードを打ってみよう。

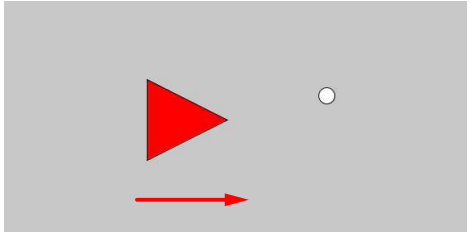


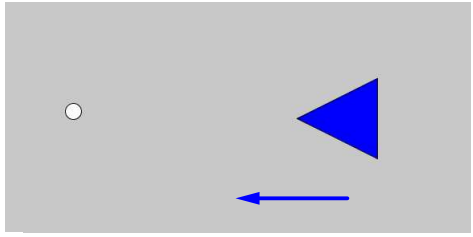

```
1  size(600, 300);
2  triangle(300, 150, 200, 200, 200, 100);
```

① 三角形を赤色にしよう。
② マウスで自由に動く直径20の白い円を書き加えてみよう。background(200);を忘れずにね。
③ 変数 xを使って、三角形をＸ座標のプラス方向へ進ませてみよう。
Ａ点のＸ座標を x にすると、Ｂ、Ｃ点のＸ座標はどうなるかな？
④ 白い円が、三角形の右にあるときは三角形は右に進み、左にあるときには三角形は青色になって左に進むようにしてみよう。

④へのヒント

白い円が右にあると三角形は右に進み、白い円が左にある三角形は左に進むよ。





１．白い円のコードの下に、右向きの三角形と左向きの三角形を用意しよう。
２．if を使って x がmouseXよりも大きいときと、小さい時に分けてコードを書いてみよう。　コピペだよ！

９ーステップ　２：if と動きを復習してみようの続き

三角形が右に進むコードだ。

```
1  int x;
2
3  void setup() {
4    size(600, 300);
5  }
6
7  void draw() {
8    background(200);
9    fill(255);
10   ellipse(mouseX, mouseY, 20, 20);
11
12   fill(255, 0, 0);
13   triangle(x, 150, x-100, 200, x-100, 100);
14   x=x+1;
15 }
```

三角形の目標となる白い円

+1 だから右に進む

右方向へ進む
右向きの三角形

上のコードは、白い円と三角形が関連付けられていないし、左に進む三角形がないね。右を見て欲しい。12～16行目を17行目にコピペして、if(mouseX>x) で白い円と三角形を関連付けたコードだ。

白い円を追いかける三角形は下のコードだ。

```
1  int x;
2
3  void setup() {
4    size(600, 300);
5  }
6
7  void draw() {
8    background(200);
9    fill(255);
10   ellipse(mouseX, mouseY, 20, 20);
11
12   if(mouseX>x) {
13   fill(255, 0, 0);
14   triangle(x, 150, x-100, 200, x-100, 100);
15   x=x+1;
16 }
17   if(mouseX<x) {
18     fill(0, 0, 255);
19   triangle(x, 150, x+100, 200, x+100, 100);
20   x=x-1;
21 }}
```

三角形の目標となる白い円

右方向へ進む右向きの赤色三角形

コピペだよ！

左方向へ進む左向きの青色三角形

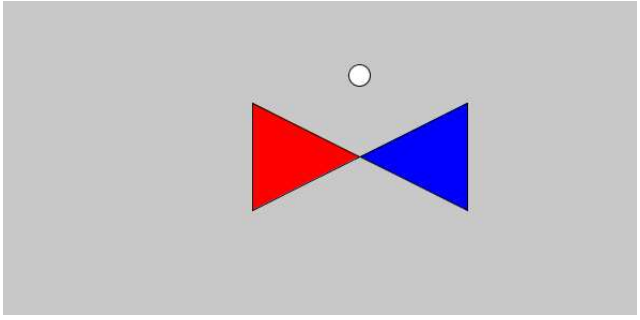
上のコードは、白い円と三角形が関連付けられて、白い円が右へ行けば三角形も右に行くようになったし、白い円が左へ行けば三角形も左へ進むね。しかし、三角形が白い円の真下か真上に来ると、三角形が消えてしまう。そして、白い円を少しでも右か左にずらすと、三角形はまた表れて進み始める。なぜだろう？
理由は左の赤字だ。

三角形が消える理由

A>B（ＡはＢより大きい）と、A<B（ＡはＢより小さい）では、ＡはＢを含まないね。
例えば、A<5 は 4 までで 5 は含まない。
A>5 も 5 は含まずに 6 から上だね。
だから A>5 , A<5 とすると A=5 がないから、 5 の行き場がなくなってしまう。
つまり、mouseX=x になったときに、コンピューターはコードの判断ができなくなってしまい、三角形が消えてしまうんだ。

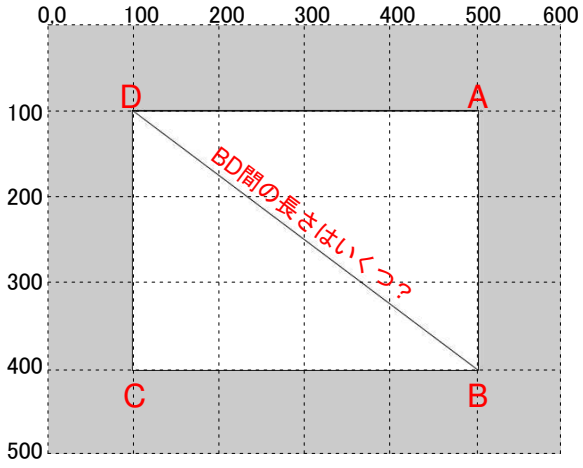
三角形を消さないための対策

A=B を加えれば良いんじゃないか！そこで、A>=B（ＡはＢ以上）か、A<=B（ＡはＢ以下）をつかって、どちらかを mouseX>=x か mouseX<=x とすれば、三角形は消えなくなるかな。
右上の１２行目の > を >= に変えて実行してみよう。



A>=B（ＡはＢ以上）と、A<=B（ＡはＢ以下）ではＡはＢを含むよね。だから mouseX=x になったときには、両方の三角形とも成り立つとコードが判断しているから、両方の三角形が表示されるんだね。

９ーステップ　３：距離と、円と円の衝突（しょうとつ）



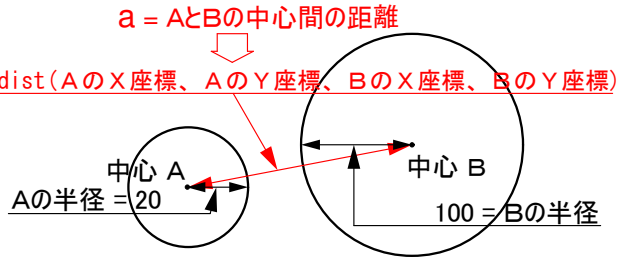
下がＢＤ間の長さを求めるコードだ。

```
1  float a;
2  void setup() {
3    size(600, 500);
4
5    rect(100, 100, 400, 300);
6    line(100, 100, 500, 400);
7    a=dist(100, 100, 500, 400);
8    println(a);
9  }
```

ウィンドウ下のコンソールに何てでた？ その数字がBD間の長さ＝距離だ。
7行目の dist は、distance(ディスタンス) 距離という意味の最初の4文字だ。つまり、dist( ); は2点間の長さ=距離を求める命令だったんだよ。それを変数 a に代入してコンソールに出力したんだ。これを使って円の衝突を考えてみよう。

円と円が衝突したら（=ぶつかったら）のコードは？

円が２つないとぶつからないね。動いたからぶつかったんだから、円は動いているはずだね。
Ｘ座標が mouseX、Ｙ座標が mouseY、直径が 40の円と、Ｘ座標が 600、Ｙ座標が 500、直径が200 の円を書いて、上のコードを下のように変更しよう。
そして、void setup( ){ } と void draw( ){ } を設定して、動かしてみよう。



Aの半径 + Bの半径 = 20 + 100 = 120

右の説明を参考にして、下のコードを書いてみよう。

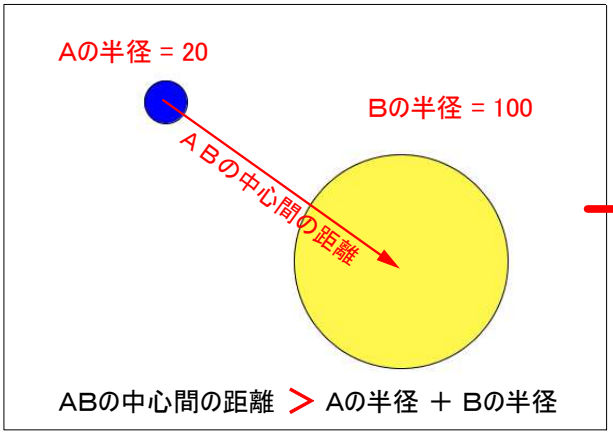
```
1  void setup() {
2    size(800, 700);
3  }
4
5  void draw() {
6    background(255);
7
8    float a = dist(                              );
9
10   fill(255, 247, 77);
11   if(a<=100+20) {
12     background(0, 255, 0);
13     fill(0, 0, 255);}
14     ellipse(                    );
15
16     fill(0, 0, 255);
17     ellipse(mouseX, mouseY, 40, 40);
18 }
```

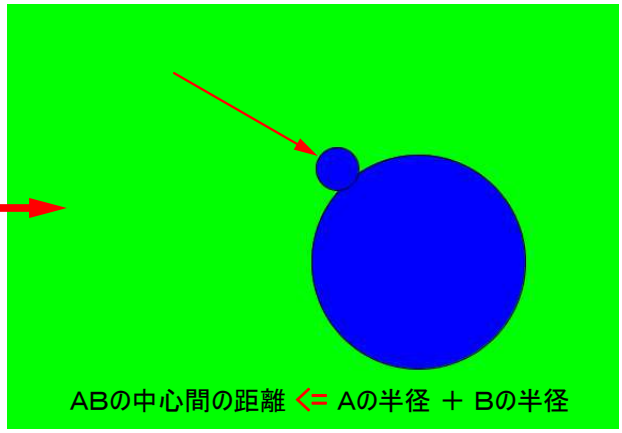
空欄にコードを書いてね
大きい円：B
小さい円：A
length-1 で保存しよう。

小さい方の円は、ellipse(mouseX, mouseY, 40, 40);でマウスにつれて動くね。大きい円は、(600, 500, 200, 200); だ。すると小さい円と大きい円の中心間の距離は、dist( );という命令で表される。dist(mouseX, mouseY, 600, 500)が中心間の距離だ。これは小数で表されるから、float a という変数で受けている。

中心間の距離 a が、円Ａと円Ｂの半径を加えた数字 120 以下になれば、２つの円はぶつかっているよね。それが左の if から下の式だ。このコードを実行すると分かるように、背景が緑色になり、大きい円が青色になると２つの円はぶつかっているんだ。

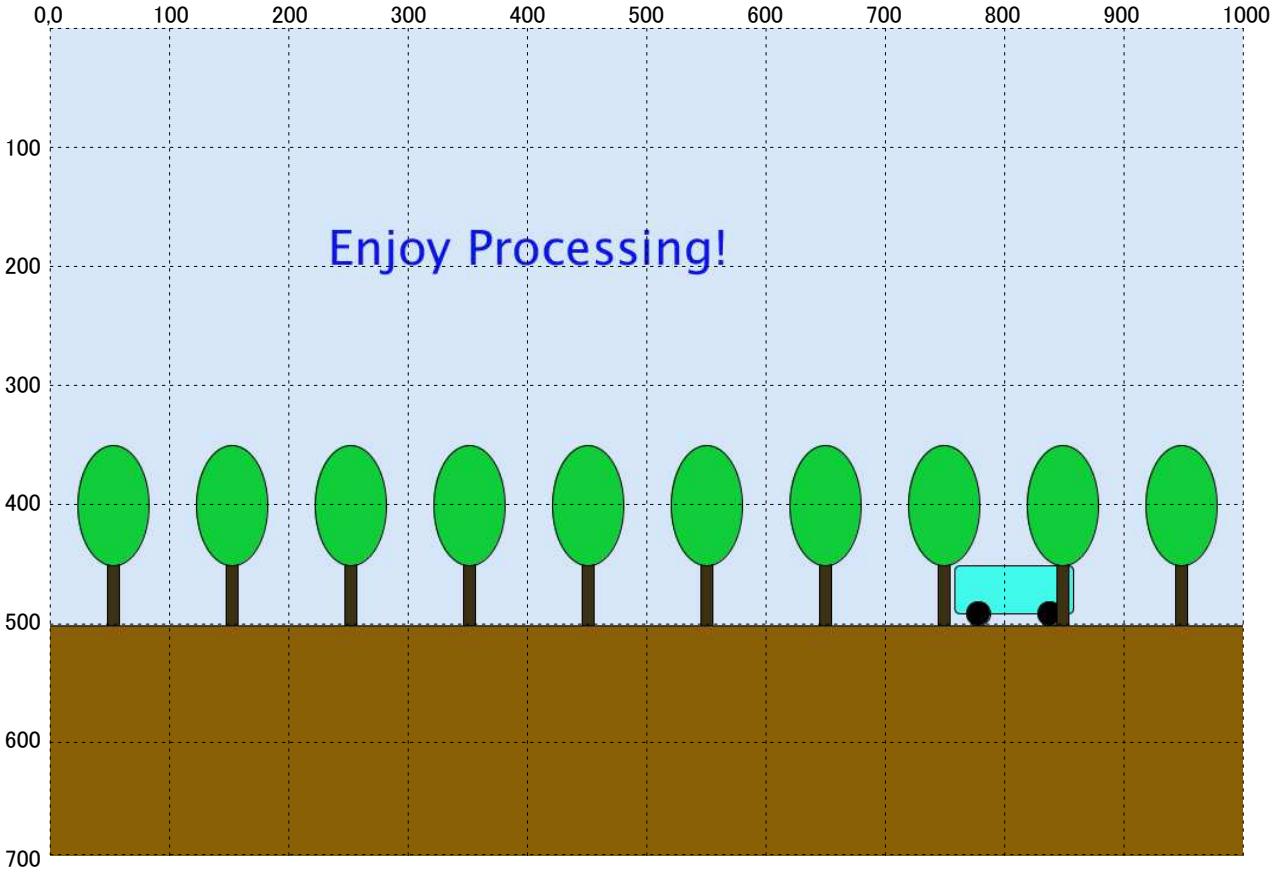
青色の小さな円が黄色の大きな円にぶつかった＝重なったので、大きな円は青色に変わり、背景は緑色に変わった。
こうしたコードは、ゲームの<当たり>などに使われるんだよ。





９ーステップ　４：中間の総合的な復習だよ

木々のむこうをバスが左から右に走っているコードを打って実行してみよう。次に、
① バスを往復させてみよう。② Enjoy Processing! の文字を右から登場させて
好きなところで止めてみよう。③自由に背景をアレンジしてみよう。



参考にしてね。

```
size(Ｘ座標, Ｙ座標);
line(Ｘ座標, Ｙ座標, Ｘ座標, Ｙ座標);
ellipse(Ｘ座標, Ｙ座標, 横幅, 縦幅);
rect(Ｘ座標, Ｙ座標, 横幅, 縦幅);
triangle(Ｘ座標, Ｙ座標, Ｘ座標, Ｙ座標, Ｘ座標, Ｙ座標);
fill(R, G, B);
noFill( );
stroke(R, G, B);
noStroke( );
strokeWeight( ？ );
background(R, G, B);
```

```
int a;　　または　int a=〇;
float a;　または　float a=〇;
for(int i=〇;　i<〇; i=i+〇または i=i-〇)
if(a<〇 または a<=〇) {　実行したいコード　　}
if(a>〇 または a>=〇) {　実行したいコード　　}
void setup(){　　}
void draw(){　　}
mouseX, mouseY
mousePressed
dist(Ｘ座標, Ｙ座標, Ｘ座標, Ｙ座標);
textSize(　？　);
text(″　　″, Ｘ座標, Ｙ座標);
frameRate( );
```

ここを見直そう。

第１回：ウィンドウの表示、図形の形の表し方
第２回：図形や背景などの色の決め方、アルファベットの表し方
第３回：変数の使い方、繰り返しの表し方
第４回：繰り返しと if を組合せた使い方
第５回：図形を移動させるには

９－ステップ　５：並木とバスのコード

```
int a=0;       //バスの行きの動き
int b=1;       //バスの帰りの動き
int c=1200;    //文字の最初の位置

void setup() {
  size(1000, 700);              //ウィンドウの大きさ
}

void draw() {
  background(212, 230, 247);    //背景の描写
  fill(139, 95, 6);             //大地の色
  rect(0, 500, width, height);  //大地の描写

  fill(12, 20, 229);            //文字の色
  textSize(40);                 //文字の大きさ
  text("                ", c, 200);   //文字を読み込む
  c=c-1;                        //文字の動き
  if(c<=330) c=330;             //文字の停止位置

  fill(66, 250, 236);           //バスの車体の色
  rect(a, 450, 100, 40, 5);     //バスの描写
  fill(0);                      //タイヤの色
  ellipse(a+20, 490, 20, 20);   //左タイヤの描写
  ellipse(a+80, 490,  ,  );     //右タイヤの描写
  a=  ;                         //バスを往復させる

  if(a>width+80) b=-1;    //バスの動き(右から左へ)
  if(a<-120) b=1;         //バスの動き(左から右へ)

  for(   i=50; i<width; i=i+100) {   //樹木の描写
    fill(62, 50, 20);                //幹の色
    rect(i, 430, 10, 70);            //幹の描写
    fill(15, 206, 57);               //枝の色
    ellipse(i+5, 400, 60, 100);      //枝の描写
}}
```

ウィンドウの外から文字を登場させるために、サイズより大きい数字を使った

大地と背景が一番奥だから、大地と背景の描写を一番先にする

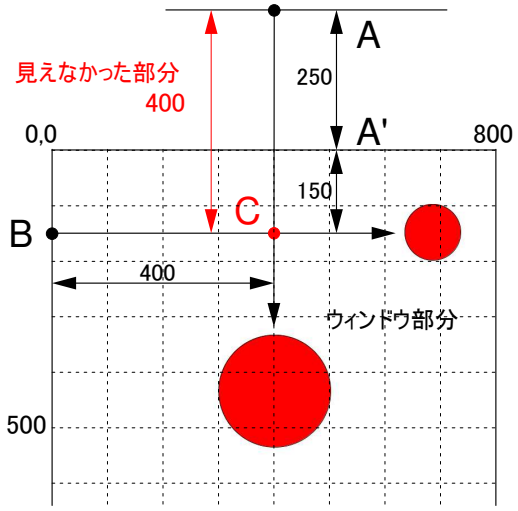
文字をX座標に平行に動かすために、X座標に変数 c を使った。
右から左へ動かすのだから、変数 c から 1 を引く。
左から 330 の位置で止める命令。

バスの車体とタイヤの色が違うので別々に指定する。
バスの車体の角を丸くするために第4の数字を使う。
タイヤはバスの車体の内側にあるので数字で調整する。

バスの車体がウィンドウの外にでてから戻るように設定。

for 文を使って、並木を繰り返し描く。
並木は一番手前なので、コードは一番最後に書く。

ステップ1　ミッションー１　③ の答え



ステップ１　ミッションー１　③ の答えの解説：

Ｃ点で表れた大きい円は、ウィンドウの外で小さい円と同じ速さで下向きに動いていたが、Ｃ点までは表示されなかった。小さい円がＸ座標を 400 まで進んだとき、if で条件が設定されたので表示されたのだ。

大きい円も小さい円も同じ速さ a=a+1; だから、同じ距離を動いたはずだね。小さい円はＣ点まで 400 動いて、大きい円も同じ 400 動いて、Ｃ点で出会った。Ｃ点のＹ座標は150 だから、隠れていた距離はＡＡ'で、400-150=250 だ。

大きい円が、小さい円と同じ距離動いたとするためには、大きい円のＹ座標から -250 しなければならないんだね。