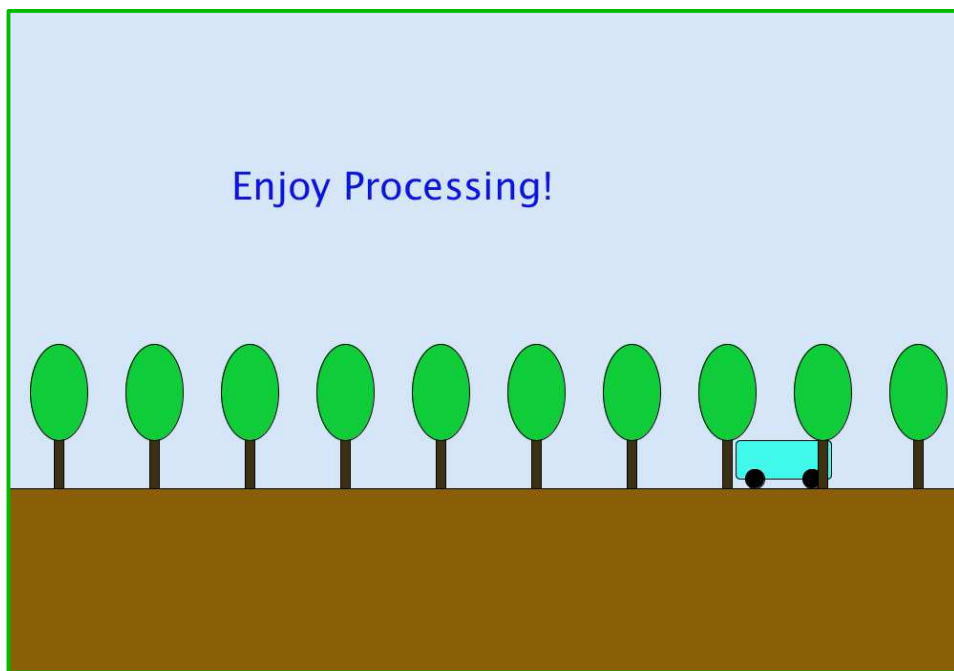


Processing

第9回



松田小学校 / 寄小学校

9 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(491, 348);
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, 180, 180);
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。jpg に注意

牛さん呼び出そう。png に注意

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えるよ。

0,0

491



348

9-ステップ 1 : if と動きを復習してみよう

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
3 void setup() {
4   size(800, 800);
5 }
6
7 void draw() {
8   ellipse(a, 150, 100, 100);
9   a=a+1;
10 }

```

実行してみよう

background();を書き加えて、黒い帯を消そう。書き加えて①~③を実行してみよう。

- ① ウィンドウの中央で赤色にしてみよう。9行目の次にもし a が中央以上なら、赤色に塗れと書くんだよ。
- 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の円の命令を書くんだ。
- 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、X座標は変数じゃないから直してね。

上のミッションをやると下のコードになるね。

```

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

```

実行してみよう

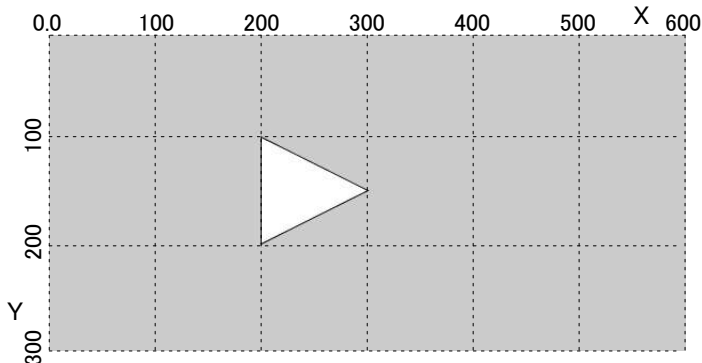
ミッションで手直したのが左下のコードだ。

- ① のコードは、11行目だ。
- ② のコードは、8行目コードを12行目にコピーで挿入して、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？ 答えはステップ4の最後にあるよ。

12行目の // を消して実行してみよう。

red-ball で保存しよう。

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);

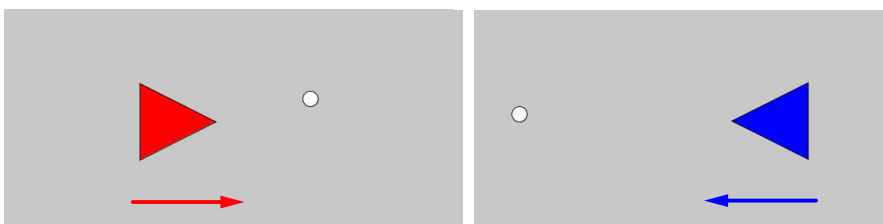
```

ミッション-2

- ① 三角形を赤色にしよう。
- ② 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。
- ③ マウスで自由に動く直径50の白い円を書いてみよう。
- 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、左にあると三角形は左に進むよ。



1. 白い円のコードの下に、右向き三角形と左向き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。 **コピーだよ!**

9 - 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

↑ 右方向へ進む三角形

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

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。

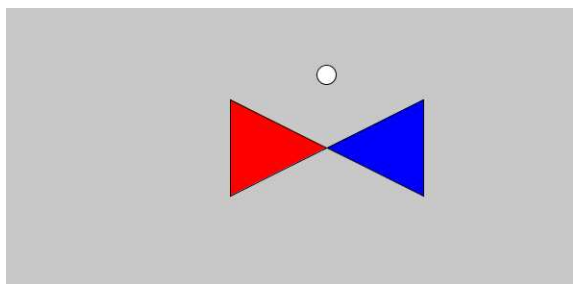
$A > 5$ も 5 は含まずに 6 から上だね。

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、 $A \geq B$ (AはB以上) か、 $A \leq B$ (AはB以下) をつかって、どちらかを $mouseX \geq x$ か $mouseX \leq x$ とすれば、三角形は消えなくなるかな。右上の12行目の $>$ を \geq に変えて実行してみよう。



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; コピペだよ!
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

↑ 右方向へ進む赤色の三角形

↑ 左方向へ進む青色の三角形

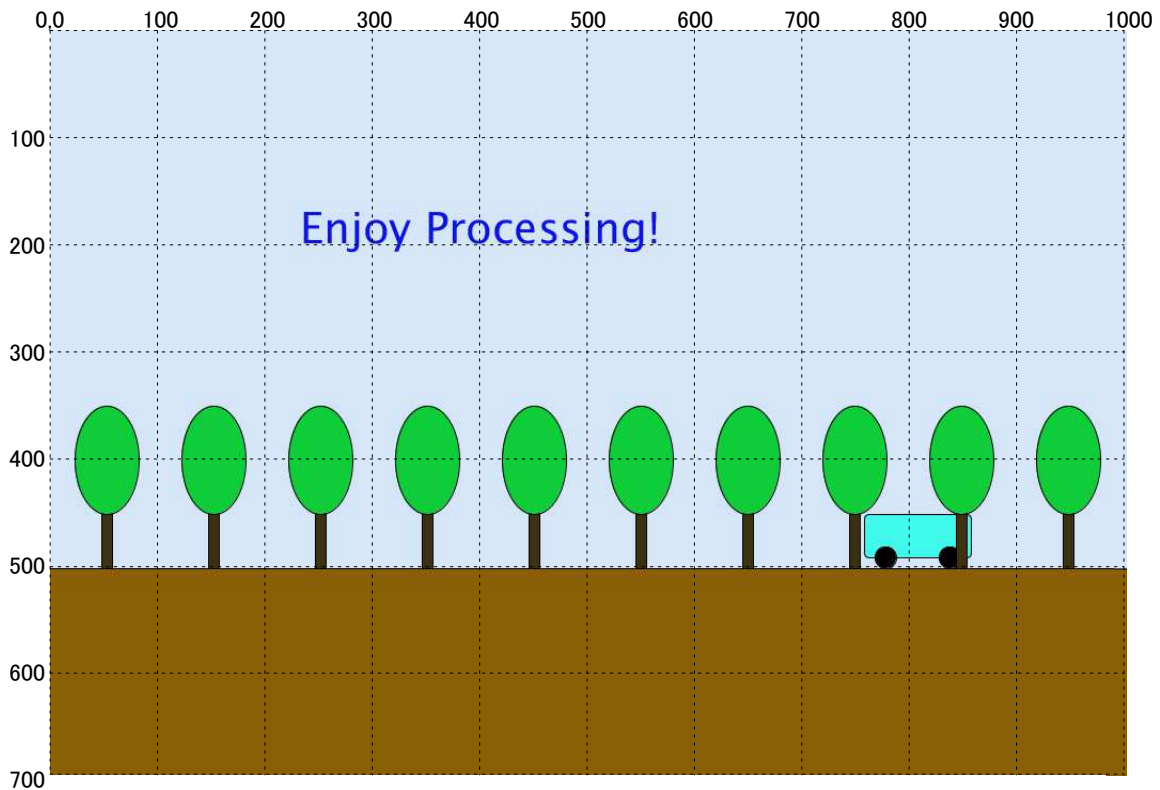
上のコードは、白い円が右へ行けば三角形も右に行くし、白い円が左へ行けば三角形も左へ進むね。しかし、三角形が白い円の真下か真上に来ると、三角形が消えてしまう。そして、白い円を少しでも右か左にずらすと、三角形はまた表れて進み始める。なぜだろう？ 理由は左の赤字だ。

$A \geq B$ (AはB以上) と、 $A \leq B$ (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-ステップ 3 : 総合的な復習だよ

木々の立ち並ぶ向こうに、バスが往復している風景を描いてみよう。

①背景を描く。②風景にバスを描き加える。③バスを動かす。④Enjoy Processing! の文字を、左から登場させて好きところで止める。



① 背景を描く。右も左も同じだから、どちらかを打ってね。

```

1 void setup() {
2   size(1000, 700);
3 }
4
5 void draw() {
6   background(212, 230, 247);
7   fill(139, 95, 6);
8   rect(0, 500, 1000, 700);
9 }

```

```

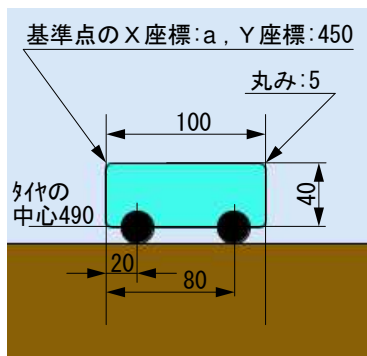
1 void setup() {
2   size(1000, 700); //ウィンドウの大きさ
3 }
4
5 void draw() {
6   background(212, 230, 247); //背景の描写
7   fill(139, 95, 6); //大地の色
8   rect(0, 500, width, height); //大地の描写
9 }

```

実行してみよう。

加える

② 風景に、バスを描き加えよう。



加える

```

1 int a=1;
2 void setup() {
3   size(1000, 700); //ウィンドウの大きさ
4 }
5 void draw() {
6   background(212, 230, 247); //背景の描写
7   fill(139, 95, 6); //大地の色
8   rect(0, 500, 1000, 700); //大地の描写
9
10  fill(66, 250, 236); //バスの車体の色
11  rect(a, 450, 100, 490); //バスの描写
12  fill(0); //タイヤの色
13  ellipse(a+20, 490, 20, 20); //左タイヤの描写
14  ellipse(a+80, 490, 20, 20); //右タイヤの描写
15 }

```

実行してみよう。

9 ステップ 4 : 総合的な復習の続きだよ

木々の立ち並ぶ向こうに、バスが左から右に走っている風景を描いてみよう。

③バスを左から右、右から左に動かしてみよう。

```
1 int a=1; //バスの右行き
2 int b=1; //バスの左行き
3
4 void setup() {
5   size(1000, 700); //ウィンドウの大きさ
6 }
7 void draw() {
8   background(212, 230, 247); //背景の描写
9   fill(139, 95, 6); //大地の色
10  rect(0, 500, 1000, 700); //大地の描写
11
12  fill(66, 250, 236); //バスの車体の色
13  rect(a, 450, 100, 40, 5); //バスの描写
14  fill(0); //タイヤの色
15  ellipse(a+20, 490, 20, 20); //左タイヤの描写
16  ellipse(a+80, 490, 20, 20); //右タイヤの描写
17  ;
18
19  if(a>width+80) b=-1; //バスの動き(右から左へ)
20  if(a<-120) b=1; //バスの動き(左から右へ)
21 }
```

加える

この部分は前と同じ

右行きと左行きの切り替え

加える

実行してみよう。

④立木を描いてみよう。

```
1 int a=1; //バスの右行き
2 int b=1; //バスの左行き
3
4 void setup() {
5   size(1000, 700); //ウィンドウの大きさ
6 }
7 void draw() {
8   background(212, 230, 247); //背景の描写
9   fill(139, 95, 6); //大地の色
10  rect(0, 500, 1000, 700); //大地の描写
11
12  fill(66, 250, 236); //バスの車体の色
13  rect(a, 450, 100, 40, 5); //バスの描写
14  fill(0); //タイヤの色
15  ellipse(a+20, 490, 20, 20); //左タイヤの描写
16  ellipse(a+80, 490, 20, 20); //右タイヤの描写
17  a=a+b;
18
19  if(a>width+80) b=-1; //バスの動き(右から左へ)
20  if(a<-120) b=1; //バスの動き(左から右へ)
21
22  for( i=50; i<width; i=i+100) { //樹木の描写
23    fill(62, 50, 20); //幹の色
24    rect(i, 430, 10, 70); //幹の描写
25    fill(15, 206, 57); //枝の色
26    ellipse(i+5, 400, 60, 100); //枝の描写
27  }}
28 }
```

この部分は前と同じ

加える

加える

実行してみよう。

9 ステップ 5 : 並木とバスのコード

```

1 int a=1; //バスの右行き
2 int b=1; //バスの左行き
3 float c=1200; //文字の最初の位置
4
5 void setup() {
6     size(1000, 700); //ウィンドウの大きさ
7 }
8 void draw() {
9     background(212, 230, 247); //背景の描写
10    fill(139, 95, 6); //大地の色
11    rect(0, 500, 1000, 700); //大地の描写
12
13    fill(66, 250, 236); //バスの車体の色
14    rect(a, 450, 100, 40, 5); //バスの描写
15    fill(0); //タイヤの色
16    ellipse(a+20, 490, 20, 20); //左タイヤの描写
17    ellipse(a+80, 490, 20, 20); //右タイヤの描写
18    a=a+b;
19
20    if(a>width+80)b=-1; //バスの動き(右から左へ)
21    if(a<-120)b=1; //バスの動き(左から右へ)
22
23    for(int i=50; i<width; i=i+100) { //樹木の描写
24        fill(62, 50, 20); //幹の色
25        rect(i, 430, 10, 70); //幹の描写
26        fill(15, 206, 57); //枝の色
27        ellipse(i+5, 400, 60, 100); //枝の描写
28
29        fill(12, 20, 229); //文字の色
30        textSize(40); //文字の大きさ
31        text("Enjoy Processing!", c, 200); //文字を読み込む
32        c=c-0.1; //ゆっくりとした文字の動き
33        if(c<=330)c=330; //文字の停止位置
34    }

```

3行目と29行目以下の文字のコードを加えてみよう。

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

この部分は前と同じ

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

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。

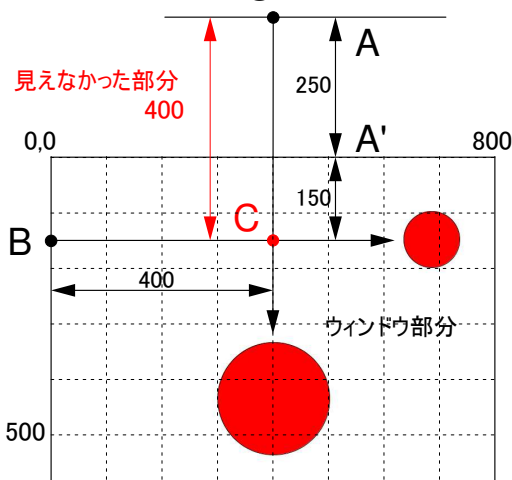
バスの車体がウィンドウの外にでてから戻るように設定。

for文を使って、並木を繰り返し描く。並木が一番手前なので、コードは一番最後に書く。

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。

実行してみよう。busで保存しよう。

ステップ1 ミッション1 ③ の答え



ステップ1 ミッション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。

1年間、よく頑張ったね。これでおしまいけど、プログラミングの
実力がついてきているから、続けて勉強していってね。

Bye-bye!