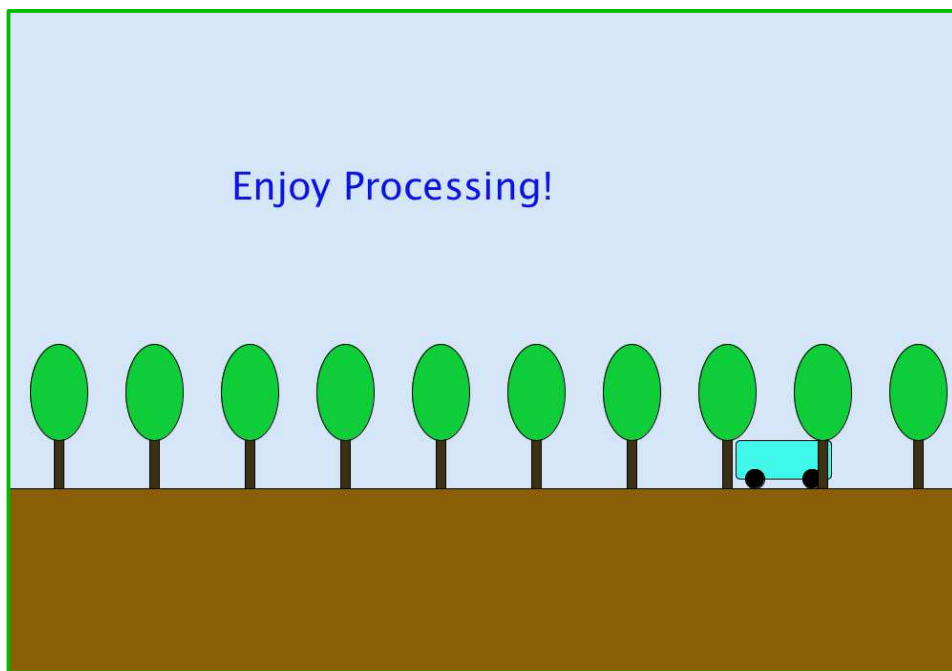


Processing

第8回



松田小学校 / 寄小学校

8 - ステップ 0 : 前回の復習だよ

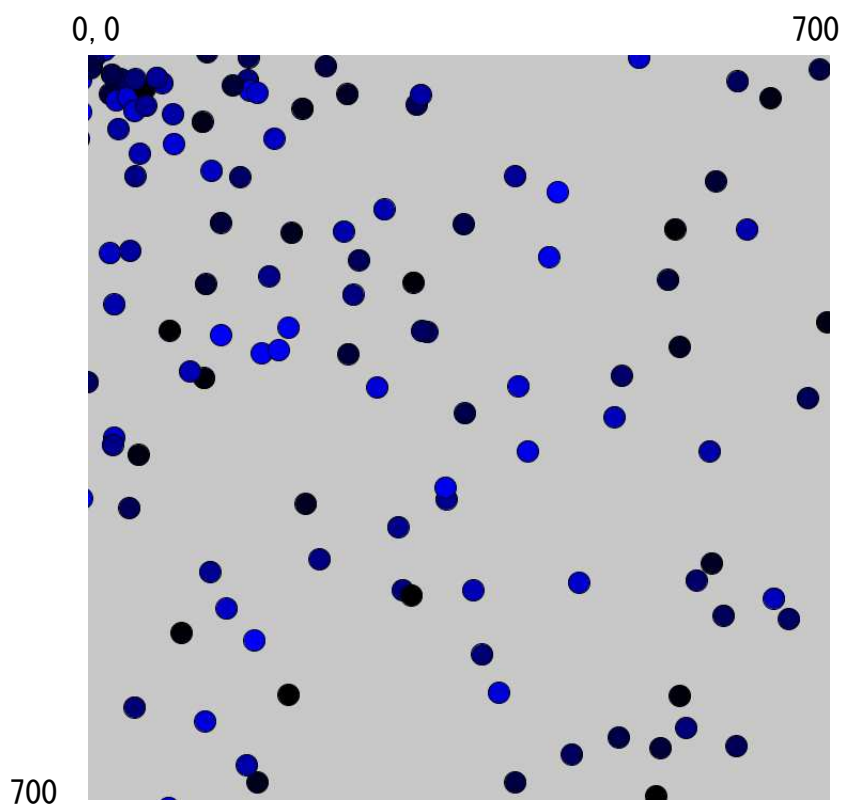
新規のファイルを開いて、下のコードを打ってみよう。

```
1 void setup() {  
2   size(700, 700);  
3   frameRate(10);  
4 }  
5  
6 void draw() {  
7   background(200);  
8   for(float a=0; a<700; a=a+0.5) {  
9     fill(0, 0, random(256));  
10    ellipse(random(-5, 5)*a, random(-5, 5)*a, 20, 20);  
11  }
```

random(-5, 5) は、-5~ 4までの数字を不規則に発生させる。

X座標とY座標は、randomをつかって -5~ 4間の数字を発生させて、それに 0 から700 までの数字 a をかけて決めている。

for() のなかに、a=a+0.5 という小数があるから、int a=0; ではなく、float a=0; なんだ。



8 ステップ 1 : 円弧(えんこ)を扱ってみよう

ファイルから新規を開いて、下のコードを打ってみよう。

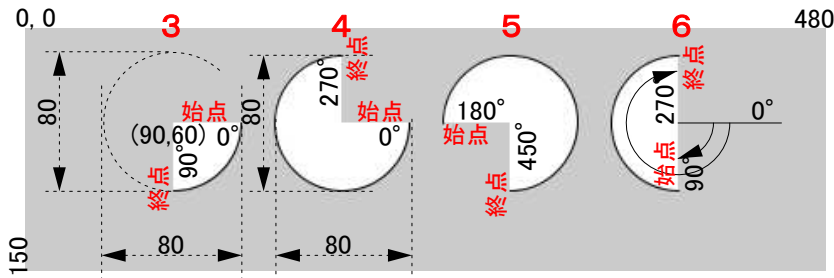
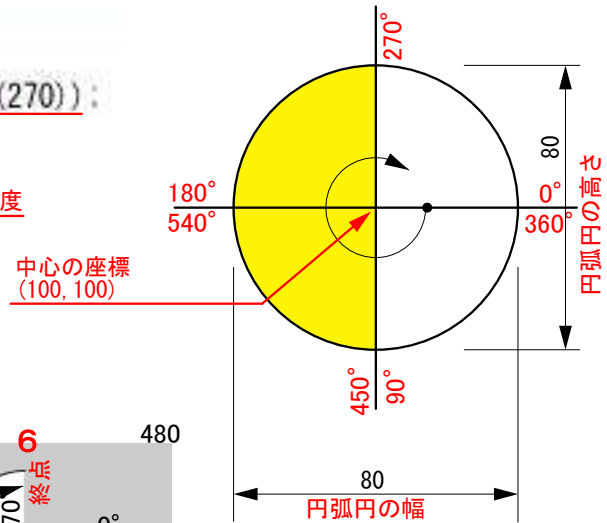
円弧(えんこ)とは、円の一部だよ。
 角度の表し方は、`radians()`を使うんだ。

`radians`(ラジアン)の角度の表し方
`radians`(赤字の数字)

```
1 size(200, 200);
2
3 arc(100, 100, 80, 80, radians(90), radians(270));
```



上のコードは、右の図の黄色い部分だ。



上のコードを下のように直して、上の図のようになるか確かめよう

```
1 size(480, 150);
2
3 arc(90, 60, 80, 80, 0, radians(90));
4 arc(190, 60, 80, 80, 0, radians(270));
5 arc(290, 60, 80, 80, radians(180), radians(450));
6 arc(390, 60, 80, 80, radians(90), radians(270));
```

コピペだよ！座標値や直しは後からね

新規のファイルを開いて、下のコードを打ってからミッションをやってみよう。ヒントをよく読んでね。

```
1 //float x;
2
3 void setup() {
4   size(700, 700);
5 }
6
7 void draw() {
8   background(100);
9   arc(350, 350, 500, 500, [ ]);
10 }
```

ミッション

- ① 左のコード1行目の `float x;` を有効にし、始点から1回転するようにコードを変えてみよう。

真上から始まり真上で終わる円弧(=円)を書いてみよう。

ヒント

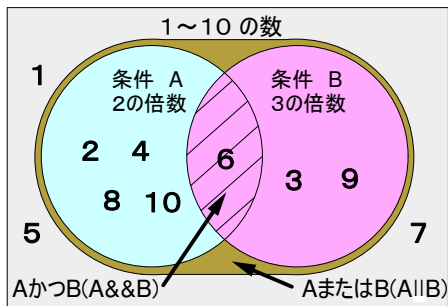
真上から始まるとは、始点が `radians(270)` ってことだね。1回転すると 360 増えるってことだから終点は `270+360` だね。 `radians(270+360)` と書けば、()内は Processing が計算してくれるよ。

動かすことは変数を、動く部分に組み込むんだよね。1回転で 360 だから、360 の部分を `x` で置き換えればOKじゃないかな。あとは、`x` が 1 ずつ増えていく式を書けば良いんだね。

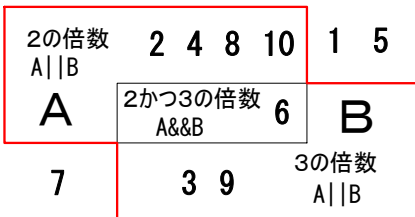
できたら、`arc-1` で保存しよう。

8 ステップ 2 : 円弧を動かそうの準備

円弧が右端に行ったら、振り向いてから進ませるようにしよう。
この先は、ちょっと難しいから事前の勉強だ！



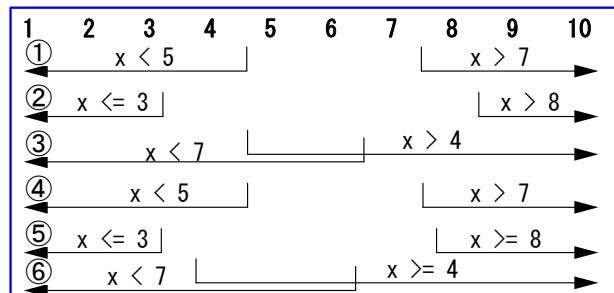
左の図で、2の倍数は条件Aにあり(=水色の部分)、3の倍数は条件B(=桃色の部分)にあるね。そして、水色と桃色が重なった部分にあるのは、**AかつB**(AとBの両方共の条件に合う)の部分にあると言うんだ。つまり、条件Aと条件Bが重なった部分にある場合だ。それを **A&&B**(呼び方はAアンドB)と表すんだよ。それに対して、水色あるいは桃色の部分にあるのは、**AまたはB**(AあるいはBの条件に合う)の部分にあると言うんだ。それを **A||B**(呼び方はAオアB)と表すんだよ。



問題① 左の図を考えてみよう

- ① Aに含まれるのは、2, 4, 6, 8, 10 だ。Bに含まれるのは？
- ② A&&B に含まれる数字は？
- ③ A||B に含まれる数字は？

ヒント



問題② 右のヒントを見て よく 考えてみよう

1から10の数のなかで、

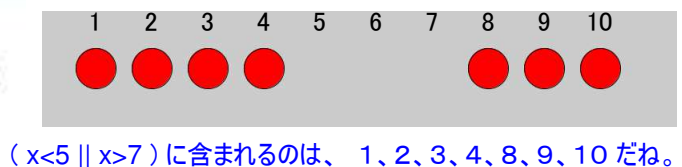
- ① if (x < 5 || x > 7) に含まれるのは、どんな数字だ？
- ② if (x <= 3 || x > 8) に含まれるのは、どんな数字だ？
- ③ if (x > 4 || x < 7) に含まれるのは、どんな数字だ？
- ④ if (x < 5 && x > 7) に含まれるのは、どんな数字だ？
- ⑤ if (x <= 3 && x >= 8) に含まれるのは、どんな数字だ？
- ⑥ if (x >= 4 && x < 7) に含まれるのは、どんな数字だ？

上の問題についてコードを書いて確認しよう。赤線の部分だけ直せばOKだよ。

```

1 size(800, 150);
2 for(int x=1; x<=10; x=x+1) [
3   if ( x<5 || x>7 ) [fill(255, 0, 0);
4   ellipse(70*x, 75, 50, 50);
5 ]
6 ]
    
```

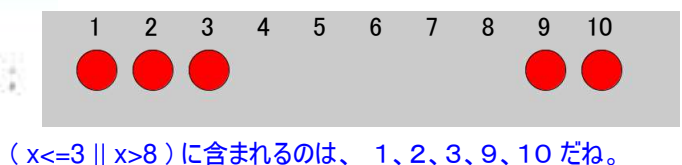
赤線部分だけが違うんだよ！



```

1 size(800, 150);
2 for(int x=1; x<=10; x=x+1) [
3   if ( x<=3 || x>8 ) [fill(255, 0, 0);
4   ellipse(70*x, 75, 50, 50);
5 ]
6 ]
    
```

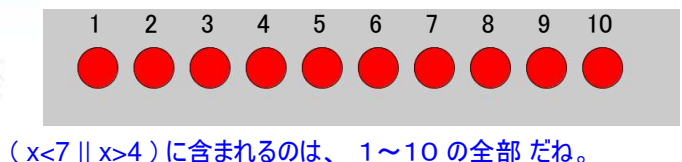
赤線部分だけが違うんだよ！



```

1 size(800, 150);
2 for(int x=1; x<=10; x=x+1) [
3   if ( x<7 || x>4 ) [fill(255, 0, 0);
4   ellipse(70*x, 75, 50, 50);
5 ]
6 ]
    
```

赤線部分だけが違うんだよ！



8 ステップ 3 : 円弧を動かそう

①~③の続き

```

④ size(800, 150);
  for(int x=1; x<=10; x=x+1) {
    if( x<5 && x>7 ) [fill(255, 0, 0);
    ellipse(70*x, 75, 50, 50);
  ]
  
```

赤線部分だけが違うんだよ！

($x < 5 \ \&\& \ x > 7$)に含まれるのは、1つもないね。

```

⑤ size(800, 150);
  for(int x=1; x<=10; x=x+1) {
    if( x<=3 && x>=8 ) [fill(255, 0, 0);
    ellipse(70*x, 75, 50, 50);
  ]
  
```

赤線部分だけが違うんだよ！

($x \leq 3 \ \&\& \ x \geq 8$)に含まれるのは、1つもないね。

```

⑥ size(800, 150);
  for(int x=1; x<=10; x=x+1) {
    if( x<7 && x>=4 ) [fill(255, 0, 0);
    ellipse(70*x, 75, 50, 50);
  ]
  
```

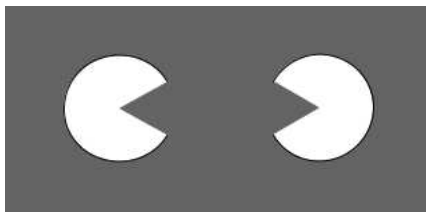
赤線部分だけが違うんだよ！

($x < 7 \ \&\& \ x \geq 4$)に含まれるのは、4、5、6だね。

①~⑥のコードは保存する必要はないよ。

新規のファイルを開き、円弧が往復するコードを書いてみよう。

行き 帰り

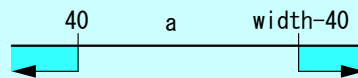


```

1 int a=39; //最初のx座標
2 int b=1; //進む方向
3
4 void setup() {
5   size(600, 200);
6 }
7
8 void draw() {
9   background(100);
10
11  a=a+b;
12  if((a<40) || (a>width-40)) { b=-b; }
13  if( b==1 ) [ arc(a, 100, 80, 80, radians(30), radians(360-30)); ]
14  if( b==-1 ) [ arc(a, 100, 80, 80, radians(180+30), radians(450+60)); ]
15 ]
  
```

13行目をコピーだよ、手直しはその後でね。

a の範囲を $a < 40$ にしたので、`int a=39;`にしないと動き始めないんだ。試しに `int a=30;`にして実行してみて。



($a < 40$) || ($a > \text{width} - 40$) だから、a が 40 未満か $\text{width} - 40$ より大きい、部分で、b が $-b$ に変わる。ここで $+ \rightarrow -$ か $- \rightarrow +$ してUターンするんだ。だから、往復するんだね。

行きでは $b == 1$ で右向きだった。帰路は $b == -1$ となると、円弧の向きが代わって左向きになるんだ。

```

12  if((a<40) || (a>width-40)) { b=-b; } ← 方向転換
13  if( b==1 ) [ arc(a, 100, 80, 80, radians(30), radians(360-30)); ] ← 右行き
14  if( b==-1 ) [ arc(a, 100, 80, 80, radians(180+30), radians(450+60)); ] ← 左行き
  
```

プロセッシングが()の中で計算してくれるから、式のままでOKだよ。

ミッション

青色で帰るようにしてみよう。2度目の行きが青色にならないようにね。

arc-4 で保存しよう。

8 ステップ 4 : 円弧を動かそうの続き

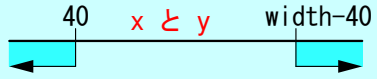
arc-4 に手直して、円弧を四周の壁で跳ね返らせてみよう。

```
1 int x=39: //最初のX座標
2 int y=39: //最初のY座標
3 int m=6: //X方向の速さ
4 int n=5: //Y方向の速さ
5
6 void setup() {
7   size(500, 500);
8 }
9
10 void draw() {
11   background(100);
12
13   x=x+m: //X座標を m ずつ飛ばしている
14   y=y+n: //Y座標を n ずつ飛ばしている
15   if ((x<40) || (x>width-40)) {m=-m;} //X方向転換
16   if ((y<40) || (y>height-40)) {n=-n;} //Y方向転換
17   if (m == 6) {arc(x, y, 80, 80, radians(30), radians(360-30));} //右行き
18   if (m == -6) {arc(x, y, 80, 80, radians(180+30), radians(450+60));} //左行き
19 }
20
```

縦横座標なので変数も2つだ。

2方向なので変数も2つだ。

arc-4 で使えるコードは、消さずに使い廻しよう。



2方向だけど、範囲指定は同じだね。X方向とY方向を、別々のif文で書いているんだ。数字などを入れ替えて、動きを確認してみよう。

X座標とY座標の両方に変数が必要だ。

arc-5 で保存しよう。

ミッション

- ① 右行きは赤色、左行きは青色にしてみよう。
- ② background(100); を fill(0,100,255,□); rect(0,0,width,height); に変えてみよう。



ヒント

- ① 右行きを赤色にするには、右行きの arc の前に fill(255, 0, 0); を入れるんだ。左行きを青色にするのは考えてね。
- ② fill の第4の数字は透明度を表すんだね。考えてみよう。

arc-6 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から、次回は復習をかねて木々の向こうをバスが走っている、そんな風景とバスのアニメーションを描いてみよう。

Bye-bye!

