

## Processing でプログラミングしよう(2)

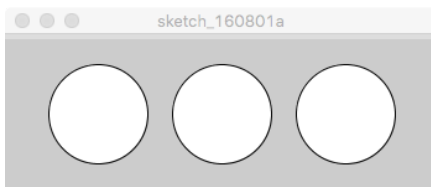
変数と for 文を用いてキャンパス上に模様を描きましょう。

### 材料(必要なもの)

Processing

#### 1 変数

変数とは、数値などを入れておく箱のようなものです。箱に名前を付けて、図形を描く命令に含まれる数値の代わりに使うことができます。変数を上手に使うと、不要な繰り返しを避けることもできます。下のスケッチとコードを確認しましょう。



```
sketch_160801a
1 size(350,120);
2 ellipse(75,60,80,80);
3 ellipse(175,60,80,80);
4 ellipse(275,60,80,80);
```

円の中心の y 軸の位置や円の横幅、高さは同じ値ですね。そこで、これらを変数に置き換えてみましょう。なお、変数を宣言するときに、変数のデータ型を指定する必要があります。データ型には、整数型の int、小数型の float、色情報型の color、文字型の char、真偽型の boolean があります。

主なデータ型

データ型	扱うデータ
int	整数
float	浮動小数点
char	文字
string	文字列
color	色
boolean	真理値

```
sketch_160801a
1 size(350,120);
2 int y = 60;
3 int k = 80;
4 ellipse(75,y,k,k);
5 ellipse(175,y,k,k);
6 ellipse(275,y,k,k);
```

#### 2 変数の計算と算術演算子

数値の代わりに変数を使って計算することもできます。例えば次のような式で計算することができます。式の中にある A というのは変数の名前です。

例  $A = A + 1;$

まず、式の「=」に注目しましょう。「=」とは等しいという意味ではなく、右辺の式の計算結果を左辺の変数 A に代入するという意味です。

次に右辺の計算に注目しましょう。A + 1は、変数Aに記憶されている値と1を足すという意味です。よって、この計算を実行する前に、次のように変数Aに値を設定しておく必要があります。

```
例 int A = 0;
    A = A + 1;
```

この計算を実行した結果、変数Aには1が代入されます。

なお、この計算で登場した+を算術演算子と用います。算術演算子には+の他にも右のようなものが用意されています。

主な算術演算子

記号	意味
+	足す
-	引く
*	掛ける
/	割る
%	剰余を求める
+=	変数 = 変数 + 数値
-=	変数 = 変数 - 数値

### 3 for 文で円を繰り返し描きましょう

先の図では、円を3つ描いていますね。コードを見ると、円を描く命令(ellipse)が3回繰り返されています。同じ命令を繰り返す場合、for文を使うと1回だけ命令するだけで同じことができるようになります。

for文の命令は次のとおりです。

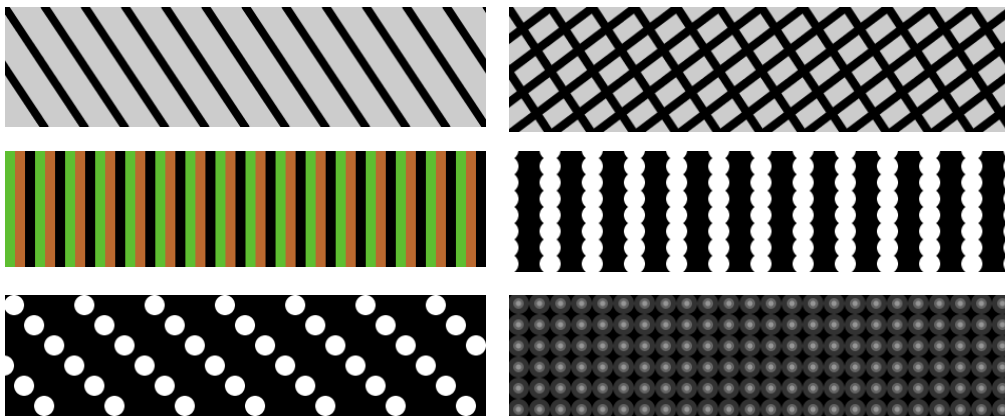
```
for(初期値 ; 繰り返す条件 ; 増分) {
    繰り返したい命令 ;
}
```

上の図とコードから、繰り返したい円のx座標の位置が、初期値は75、次が175、最後が275と100ずつ増える(増分)こと、繰り返す条件となるxの値の上限がわかりますね。

```
for_circle
1 size(400,280);
2 background(0);
3 int x;
4 int y;
5 for(x=20;x<=width;x+=40){
6     for(y=20;y<=height;y+=40){
7         ellipse(x,y,40,40);
8     }
9 }
```

### 4 問題演習

for文を用いて次のような模様を考えて描きましょう。



#### コツ(留意点)

繰り返す際に、初期値・繰り返す条件・増分がいくつかしっかりと考えて変数を使うとうまくいくでしょう。

#### 作成者

北海道帯広緑陽高等学校 前田健太郎  
k\_maeda@hokkaido-c.ed.jp