

# Processingでプログラミングに挑戦!

## —第2回 変数を使ってみよう—

杉浦 学

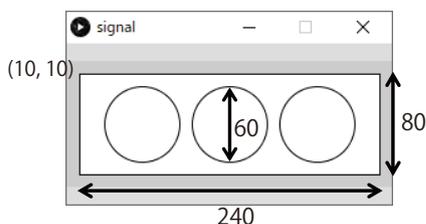
鎌倉女子大学

### 宿題の解説

前号では、図形を組み合わせて信号機 (図-1) を描く宿題を出題しました。

#### ● 前号の宿題

お手本を参考に、信号機を描いてみましょう。今回はモノクロですが、次号で色を付けていく予定です。



画面のサイズは横260, 縦100  
円の中心は左から(60, 50)(130, 50)(200, 50)

図-1 信号機のお手本

この宿題の解答例をスケッチ 1 に示します。信号機の部品は 4 つ (枠 1 つと灯火 3 つ) なので、ウィンドウの大きさを指定する size 関数と合わせ、5 つの命令があれば十分です。解答例では、5 つの命令を連続で書くだけでなく、コメントと空行を使って全体を 3 つに分割し、それぞれの部分の役割を分かりやすくしてみました。スケッチの最初の行に /\* 信号機を描くスケッチ \*/ のようなタイトルをコメントとして書いておくのもよいでしょう。

練習のための短いスケッチであっても、コメントや空行を使って分かりやすく書いておくことは重要です。しばらくたってから自分が書いたスケッチを見返して、自分で改造する場合にも役立つからです。

```
//描画の準備
size(260,100);

//枠を描く
rect(10,10,240,80);

//3つの灯火を描く
ellipse(60,50,60,60);
ellipse(130,50,60,60);
ellipse(200,50,60,60);
```

スケッチ 1 信号機を描く (宿題の解答例)

### 塗りつぶし

宿題の信号機に色を付けてみましょう。fill 関数を使うと、図形の塗りつぶしの色を指定することができます。色の指定は赤 (Red)、緑 (Green)、青 (Blue) の 3 色 (RGB と呼ぶことがあります) のそれぞれが、どの程度含まれているかを 0 から 255 の数値で指定します。色の指定には、灰色の明るさを 1 つの数値で表現したグレースケールを使うこともできます。この場合は、0 が黒で 255 が白になり、128 は白と黒の中間の灰色となります。fill 関数で塗りつぶしの色を一度指定すると、その後で描いた図形はすべて指定された色で塗りつぶされることに注意する必要があります。

この fill 関数を使って、宿題の信号機に色を付けたものを図-2 とスケッチ 2 に示します。色の指定のための命令を追加したので、スケッチ 1 の状態か

らコメントや空行の入れ方なども変更してあります。

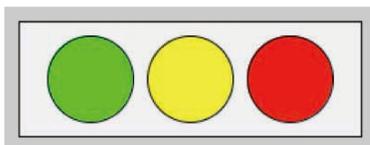


図-2 色をつけた信号機

```
//描画の準備
size(260,100);

//薄い灰色の枠を描く
fill(240);
rect(10,10,240,80);

//青信号(実際は緑色)の灯火を描く
fill(0,255,0);
ellipse(60,50,60,60);

//黄信号の灯火を描く
fill(255,255,0);
ellipse(130,50,60,60);

//赤信号の灯火を描く
fill(255,0,0);
ellipse(200,50,60,60);
```

スケッチ 2 色をつけた信号機を描く

スケッチの中で使う RGB の値を確認するには、エディタのツールメニューから「色選択」をクリックして表示されるカラーパレットが便利です (図-3)。パレットを調整して好きな色を選択すると、RGB の値を確認することができます。



図-3 色選択からカラーパレットが利用できる

なお、塗りつぶしをなくしたい場合は、noFill関数を使います。このnoFill関数のような命令は、動作を指定する数値や文字が不要なため、パラメータを指定するカッコの中には何も書きません。

## ■ 事前に手順を整理する

単純なスケッチであれば、一息に最初から最後まで書いてしまうのも大変ではありません。しかし、スケッチが複雑になってくると、「何をどのような順番ですべきか」をあらかじめ整理しておく方がうまくいきます。1つの方法として、大まかな手順を最初にコメントとして記述し、その後で具体的な命令をそのコメントの下に書いていくという方法が有効です。スケッチ 2 を作り始める前に、スケッチ 3 のように手順をコメントで整理したスケッチを用意します (この時点でスケッチを実行しても何も図形は描かれませんが)。手順が整理できたところで、それぞれのコメントの下の行に具体的な Processing の命令を順番に追加していきます。命令を少し追加したらこまめに実行をし、結果が意図どおりかを確かめながら進めるとよいでしょう。こうした手順を踏むことで、できあがったスケッチは自然と構造が整理されて、適切なコメントが挿入されたものになります。

```
//描画の準備

//薄い灰色の枠を描く

//青信号(実際は緑色)の灯火を描く

//黄信号の灯火を描く

//赤信号の灯火を描く
```

スケッチ 3 スケッチ 2 の下書き



手順を事前に考えて整理しておく訓練は、プログラミングだけでなく「試験勉強やイベントの計画を立てる」「文章の章立てを考える」「慣れていない仕事の作業効率を改善する」といったさまざまなことに応用ができるでしょう。

## 輪郭線

図形の輪郭などの線の太さの初期値は1ピクセルに設定されていますが、これを変更するには strokeWeight 関数を使います。カッコの中に指定するパラメータは1つで、太さを数値で指定します。

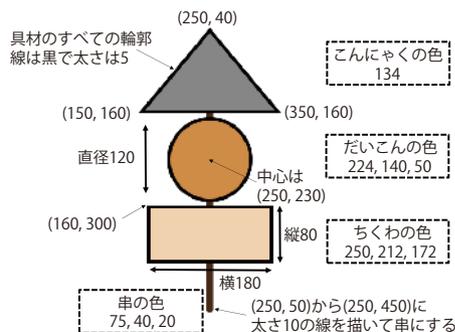
輪郭線の色を指定するには、stroke 関数を使います。fill 関数と同じように、カッコの中には RGB の3つの数値か、グレースケールの場合は0から255の数値を1つ指定します。

輪郭線をなしにしたい場合は、noStroke 関数を使います。noFill 関数と同じようにパラメータを指定するカッコの中には何も書きません。

## □ 練習問題

以下のお手本を参考に「おでん」のイラストを描いてみましょう。スケッチ3のように、最初にコメントで串や具材などの部品を描く流れを日本語で整理してから始めましょう。

ウィンドウの大きさは縦横500にして、背景を白で塗りつぶします。背景の色を設定するには、background 関数を使います。串はこれまでに解説した、strokeWeight 関数、stroke 関数を使います。基本的な図形の描画については、前号で解説しました。



## <解説>

### おでんを描くスケッチの例

```
//描画の準備
size(500,500);
background(255);//背景は白

//串を描く
strokeWeight(10);//太さは10
stroke(75,40,20);//色は茶色
line(250,50,250,450);

//具材に共通の輪郭線を設定
strokeWeight(5);//太さは5
stroke(0);//色は黒

//こんにゃくを描く
fill(134);//色は灰色
triangle(250,40,350,160,150,160);

//だいこんを描く
fill(224,140,50);//色は濃い茶色
ellipse(250,230,120,120);

//ちくわを描く
fill(250,212,172);//色は薄い茶色
rect(160,300,180,80);
```

## 変数の宣言・代入・参照

信号機のデザインを図-4のように変更したい場合について、考えてみることにしましょう。

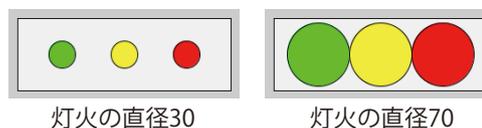


図-4 信号機のデザイン変更

スケッチ2の内容をよく検討してみると、灯火の直径を表す60という数字が6回書かれています。灯火の大きさを調整したい場合は、青黄赤のそれぞれの灯火を描いている ellipse 関数のパラメータを変更する必要があります。具体的な変更箇所は6カ所になります。このような変更の手間を減らす1つの方法が、「変数」を使うことです。

スケッチの中で使う値をコンピュータのメモリに保存しておき、スケッチの中で利用できるようにするのが変数の役割です。変数には分かりやすい名前をつけておきます。灯火の直径を d という変数（直径 diameter の頭文字）に保存して利用するスケッチの例を見てみましょう。今回のスケッチから左に行番号を付けてあります。

1	//描画の準備
2	size(260,100);
3	
4	//灯火の直径 d(diameter)
5	int d;
6	d = 60;
7	
8	//薄い灰色の枠を描く
9	fill(240);
10	rect(10,10,240,80);
11	
12	//青信号(実際は緑)の灯火を描く
13	fill(0,255,0);
14	ellipse(60,50,d,d);
15	
16	//黄信号の灯火を描く
17	fill(255,255,0);
18	ellipse(130,50,d,d);
19	
20	//赤信号の灯火を描く
21	fill(255,0,0);
22	ellipse(200,50,d,d);

スケッチ4 変数を使った信号機

変数を使う場合は最初に1回だけ「宣言」をする必要があります(「定義」という場合もあります)。ス

ケッチ4の5行目で、整数を保存するための変数 d を宣言しています。宣言の際には、変数に保存できるデータの種類である「データ型」を指定します。Processingで利用できるデータ型はたくさんありますが、入門の段階では、int 型(整数)と float 型(小数点のついた数)の2つが使えるようになれば十分です。宣言の際にはデータ型(この場合は int)を先頭に書き、半角のスペースを挿入したあとに変数の名前(この場合は d)を書きます。他の命令と同じように最後のセミコロンも忘れないようにしましょう。

変数に値を保存することを、「代入」と呼びます。6行目では変数 d に60という値を代入しています。スケッチ4では宣言と代入を別の行で行っていますが、int d = 60; のように、宣言と代入を同時に行うこともできます。

変数の値を読み出すことを、変数を「参照」と言います。14・18・22行目では具体的な数値をパラメータに指定せず、d に代入してある変数の値(スケッチ4の例では60)を読み出し、その値を直径にした円(灯火)を描いています。具体的な値の代わりに変数の名前(この場合は d)を書けばよいということです。

スケッチ4のようにしておけば、3つの灯火の直径を変更する場合は、6行目で d に代入している数値の1カ所を変更するだけで済むことになります。実際に60の数値を変更して、図-4のようなデザインの調整が手間なくできることを確認してみましょう。

## 変数と計算

Processingで使える変数は、スケッチ4の変数 d のように自分で宣言をしたものに加えて、宣言をしなくても使えるものがあります。その代表例がウィンドウの幅と高さが保存されている width と height です。スケッチの最初に size 関数で指定したウィンドウの幅と高さはこれらの変数に保存されて、スケッチの中で参照することができます。



スケッチ5ではウィンドウのサイズをsize関数でどのように指定しても、四隅を結んで×印を描くことができます(図-5)。2行目のline関数では、ウィンドウの左上の角(0, 0)から右下の角(480, 120)に直線を引きます。3行目では、ウィンドウの右上の角(480, 0)から左下の角(0, 120)に直線を引きます。ここで示した座標はウィンドウの横幅が480、高さが120の場合のものですが、直線の始点と終点の座標はウィンドウの大きさによって変わってきます。具体的な座標を書かずに、変数widthと変数heightを使うことで、さまざまな場合に対応できるスケッチになります。

```
1 size(480,120);
2 line(0,0,width,height);//左上から右下へ
3 line(width,0,0,height);//右上から左下へ
```

スケッチ5 ウィンドウのサイズに応じて×印を描く

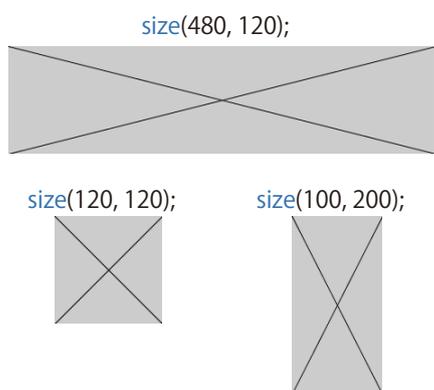


図-5 ウィンドウのサイズと連動する×印

スケッチ5の最後に新しい命令を加えたものがスケッチ6です。実際にスケッチを書いて結果を試してください。4行目の命令にある/は割り算(普段は÷と書くことが多いと思います)を指示する「演算子」と呼ばれる記号です。width/2でウィンドウの横方向の中心を、width/2でウィンドウの縦方向の中心を計算し、それを円の中心の座標としています。

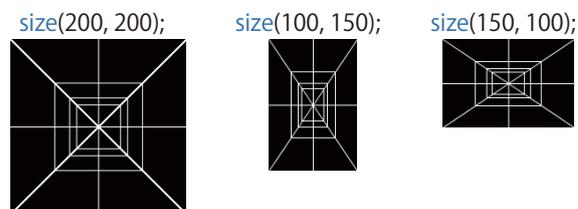
```
1 size(480,120);
2 line(0,0,width,height);//左上から右下へ
3 line(width,0,0,height);//右上から左下へ
4 ellipse(width/2,height/2,60,60);
```

スケッチ6 ×印と円を描く(文献1) P.43より

### ● 宿題

スケッチ6を応用し、お手本のようなクモの巣の模様を描くスケッチを書いてみましょう。size関数で指定したウィンドウのサイズに合わせて形が変化するよう、widthとheightの変数と割り算を使って作ります。

ヒント:rect関数は初期状態では四角の左上の座標をパラメータで指定します。これを四角の中心に変更するためにはrectMode(CENTER);という命令をrect関数の前に書いておきます。



四角形は小さい順に、画面の縦・横の長さのそれぞれの四分の一、三分の一、二分の一になるように割り算で計算します

### 参考文献

- 1) Reas, C., Fry, B. 著, 船田 巧 訳: Processingをはじめよう 第2版, オライリー・ジャパン(2016).

(2019年11月16日受付)

杉浦 学 (正会員) manabu@kamakura-u.ac.jp

鎌倉女子大学家政学部家政保健学科准教授。慶應義塾大学大学院政策・メディア研究科後期博士課程修了。博士(政策・メディア)。プログラミング教育をはじめとした情報教育に関する研究に取り組む。中高生向けの著書に『Scratchをはじめよう! プログラミング入門 Scratch3.0版』(日経BP社)など。