

# Processing

第3回



松田小学校／寄小学校

### 3 ステップ 0 : 前回の復習だよ

次のコードを打ってみよう。どんな図形になるかな？

```
1 size(600, 400);  
2 fill(0, 0, 255);  
3 rect(150, 100, 300, 200);
```

#### ミッション

2行目のfill( )の数字を変えて好きな色にしてみよう。ツール→色選択を使ってね。

上のコードに次の2行目を加えて実行してみよう

```
5 textSize(50);  
6 text("xxxxxxx", 200, 200);
```

← xxxxxx には、あなたの名前をローマ字で入れてね。

#### ミッション

あれ！あなたの名前が見えないね。どうしたら見えるようになるだろうか。  
何かコードを加えて、見えるようにしてみよう。

0.0

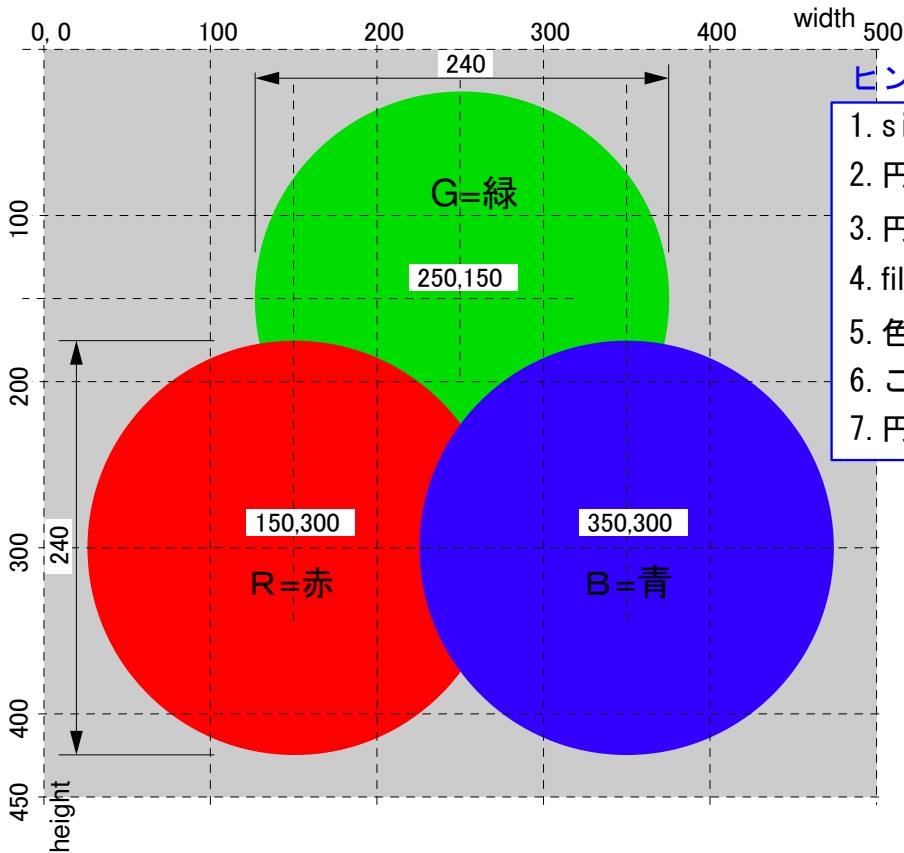
600

400



### 3ステップ 1 : アルファベットを使ってみよう

ファイルから新規を開いて、下図のコードを書いてみよう。



#### ヒント

1. size はいくつかな？
2. 円は ellipse だったね。
3. 円の中心はどこだ？
4. fill は図形毎に指定するんだよ。
5. 色決めは色選択を使うんだ。
6. この円には縁取りがないね。
7. 円の大きさは全部同じだよ。

直径を 240 から 260 に変更したい。  
そこで登場するのが変数 a だ。

1. `int a;` ← まとめて
2. `a= 240;` `int a=240;` でもOK
- 3.
4. `size (500, 450);`
5. `noStroke ();`
- 6.
7. `fill (0, 255, 0);`
8. `ellipse (250, 150, a, a);`
- 9.
10. `fill (255, 0, 0);`
11. `ellipse (150, 300, a, a);`
- 12.
13. `fill (0, 0, 255);`
14. `ellipse (350, 300, a, a);`

#### 変数の使い方

一番最初にく `a` を使いますよ、と宣言するんだ。しかも、`<a>` は `int` つまり整数だと限定する。もし、小数なら `float` を使うんだ。  
`int` は integer(整数という意味)の省略形だよ。

この宣言さえすれば、整数なら `a` にいくつを入れてもいいんだ。  
`240` を `260` に変更したり、`100` や `200` に変えて、どう変わるか確かめてほしい。

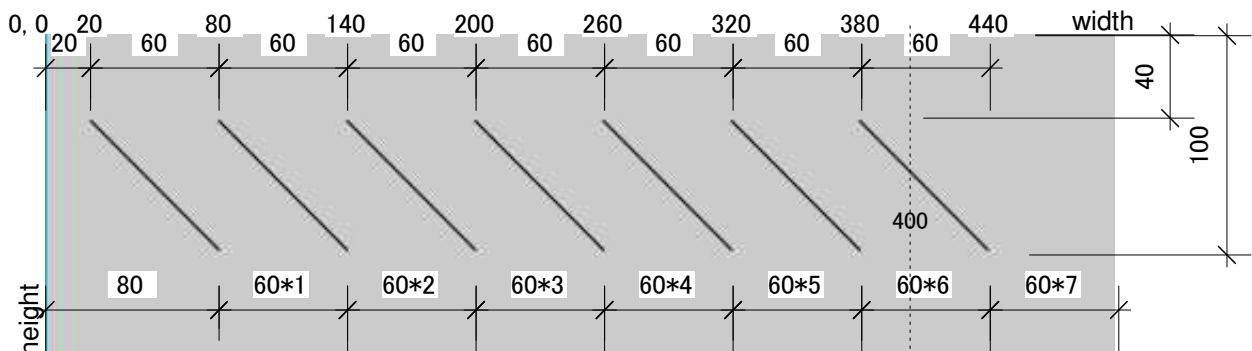
`a` のような文字を変数というんだけど、`a` じゃなくて、どんな文字でもいいんだ。  
`b` でもいいし、`baka` でもいいんだよ。  
また、変数は必要に応じて、何種類使ってもいいんだ。

### 3ステップ 2 : 変数を使いこなそう

ファイルから新規を開いて、下図のコードを書いてみよう。

```
1 size(500,150);
2
3 line(20,40,80,100);
4 line(80,40,140,100);
5 line(140,40,200,100);
6 line(200,40,260,100);
7 line(260,40,320,100);
8 line(320,40,380,100);
9 line(380,40,440,100);
10
```

似ているコードを書くときは、**範囲指定→コピー→貼り付け(=コピー)**をして、部分的に訂正していけば良いんだ。左のプログラムは、line も同じだし数字も2ヶ所直すだけだ。こんなときは**コピー**を使おう。  
マウスの左ボタンを押したまま、右か左に滑らすと範囲指定ができるよ。マウスを目的の所へ移動したら、右クリックして貼り付けを選ぼう。2行、3行まとめて**コピー**すると早いね。



#### 変数で対応できるか

上のコードを実行すると、上のような図形になるね。7本の直線だから、1本ずつ書いても良いけれど、100本、1000本となると、ちょっと無理だね。そこで、変数を使ってみよう。上のコードに変数の宣言をしよう。

```
1 int a=100; ← 変数 a の宣言
2 size(500,150);
3
4 line(20,40,80,a);
5 line(80,40,140,a);
6 line(140,40,200,a);
7 line(200,40,260,a);
8 line(260,40,320,a);
9 line(320,40,380,a);
10 line(380,40,440,a);
11
```

**変数じゃない！！**  
変数を使っても、コードの行数は変わらないね。これでは意味がないから、変数をつかう場合じゃないね。  
左のコードには何か規則性がないだろうか？  
最初の数字は、60 ずつ大きくなっている。  
2番目の数字は、変わらないね。  
3番目の数字は、60 ずつ大きくなっている。  
4番目の数字(a)は、変わらないね。  
こんな場合は、下のような **for** をつかうんだ。

60 ずつ大きくなっている。

始点のX座標 20は、60 ずつ大きくなっているのだから、10行目では380となるね。その時の終点のX座標は 440 で、400の制限を超えているけど、どちらかが制限内ならOKなんだ。

```
1 size(500,150);
2
3 for(int a=20; a <= 400; a=a+60) {
4     line(a,40,a+60,100);
5 }
```

このコードの意味は次を見てね。

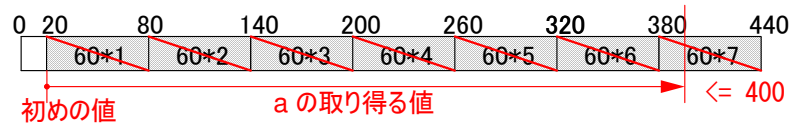
### 3-ステップ 3 : 繰り返しをもっと深く...

- 1 `size(500,150);` の意味は判るよね。
- 3 `for (int a=20; a<=400; a=a+60) { line (a, 40, a+60, 80); }` は下の意味だよ。

```
for          //繰り返しの開始記号
(int a=20;   //変数 a を20(初期値)から、
a<=400;    //変数 a が 400 になるまで
a=a+60)    //変数 a を 60 ずつ大きくし、
{line(a, 40, a+60, 80);} //{ }内を変化させながら描く
```

下の図を見て欲しい。  
a は 400 以下と制限されている。380 を超えると、次は 440 となるけれど始点が終点のどちらかが、400以下であれば、7本目まで描けるのだ。

ファイルから新規を開いて上のコードを書いてみよう。



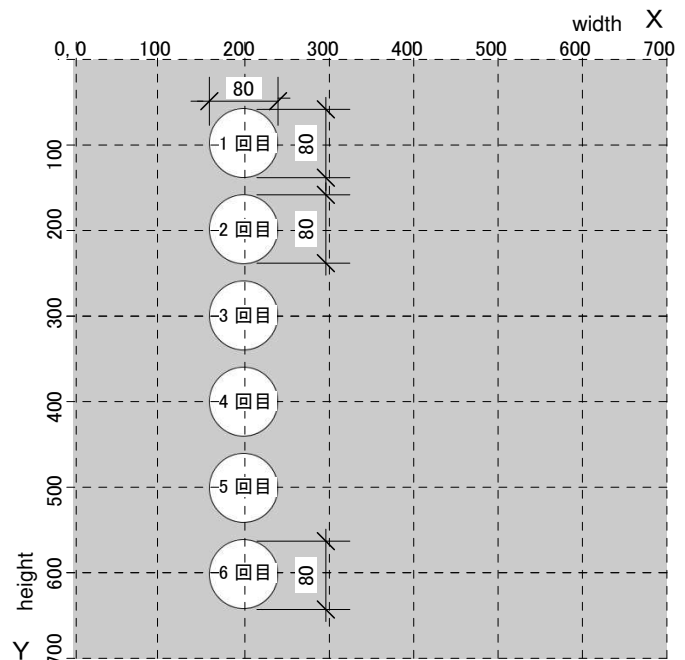
もし、整数以下の細かさで書くときは、`for(float a=1; a<=5; a=a+1.5)` となるよ。

上のコードを実行したら、  
下のコードに手直して実行してみよう。

```
1 size(700,700);
3 for(int a=1; a <= 6; a=a+1)
  {ellipse(200, 100*a, 80, 80);}
```

for の意味は、ellipse を 1 から 6 まで  
繰り返せということだね。for の中身を  
くわしく書くと、下のようになるね。

```
1 回目 ellipse(200, 100*1, 80, 80);
2 回目 ellipse(200, 100*2, 80, 80);
3 回目 ellipse(200, 100*3, 80, 80);
4 回目 ellipse(200, 100*4, 80, 80);
5 回目 ellipse(200, 100*5, 80, 80);
6 回目 ellipse(200, 100*6, 80, 80);
```



#### ミッション-1

1. 上のコードの下に、最初の円の中心座標が X=100, Y=200 で始まる直径 50 の円を X方向に6ヶ描いてみよう。これは保存しなくてもOKだよ。

#### ミッション-2 ファイル→新規

1. `size(700, 700);` で、最初の円の座標が X=100, Y=100で始まる直径 80 の円を斜め 45° 方向に6ヶ描いてみよう。
2. 円に好きな色を付けてみよう。
3. for-1 と名前を付けて保存しよう。

#### for に対する色の付け方

- ① for の前に fill を書く。
  - ② for の後の { } 中の ellipse の前に書く。
- 実行結果は同じだけど、①と②の意味の違いを考えて。

```
1 size(700, 200);
2
3 fill(255, 0, 0);
4 for(int a=1; a<=6; a=a+1){
5   ellipse(100*a, 100, 80, 80);
6 }
7
8
9 for(int a=1; a<=6; a=a+1){
10  fill(255, 0, 0);
11  ellipse(100*a, 100, 80, 80);
12 }
```

# 3-ステップ 4 : 数式をやっつけよう

ファイルから新規を開いて、下のコードを書いてみよう。

```
1. println(5+8);      6. println(5%3);
2. println(5-8);      7. println(5+3*2);
3. println(356*26);   8. println((5+3)*2);
4. println(756/27);   9. println(25-17/3);
5. println(5/3);      10. println(25-17%3);
```

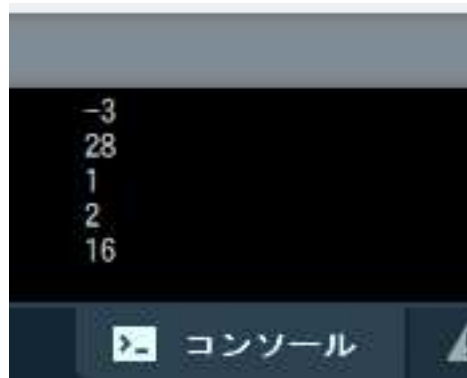
```
1 println(5-8);
2 println(756/27);
3 println(5/3);
4 println(5%3);
5 println((5+3)*2);
6
7
```

小さな数字から大きな数字を引くとマイナスになる。  
 割り切れたときは商を返してくる  
 割り切れないときは整数部分だけ返してくる  
 % は割った余りだけを返してくる  
 カッコがあるときはカッコの中から先に計算する  
 保存は不要だよ。

計算結果は下のコンソールにできるよ

1. 3. 7. 9. 10.  
は各自で確認してね。

println( ); はコンソールに計算結果をだす命令だよ。



### プロセッシングで使う比較記号

- = 等しい(等号)      ★ 注意
- != 等しくない      ≠ じゃないよ!
- A < B    AはBより小さい
- A > B    AはBより大きい
- A <= B   AはB以下(等しいか小さい)
- A >= B   AはB以上(等しいか大きい)

プログラミングでは = は等号ではなく、代入記号だよ。右辺の数字や記号を、左辺に代入する意味に使うんだよ。

### プロセッシングで使う計算記号

- + 足す
- 引く
- \* 掛ける
- / 割る
- % 余り

たとえば、7/5=1...2 だ、5/5=1...0 だから、2/5=0...2 だ。% は余りだから、2%5=2 5%5=0 となるよ。println( ); で確認しよう。

< と > は、たとえば、a < 5 は4までで、5は含まないんだ。  
<= と >= は、たとえば、a <= 5 は5までで、5を含むんだ。  
< や > の後ろに = が付くと、付かないでは意味が違うんだ。  
この違いは大切だから、頭にインプットしてね。

i = 0 のとき、i = i + 1 の意味は下の通りだよ。

ファイルから新規を開いて、下のコードを書いてみよう。

```
1. size(500, 300);
2.
3. for(int i = 0; i <= 3; i=i+1){
4.   noStroke();
5.   fill(255, 0, 0);
6.   rect(80+100*i, 100, 50, (60+i*10));
7. }
```

- 1回目    i = 0
- 2回目    i = 0+1 = 1
- 3回目    i = 1+1 = 2
- 4回目    i = 2+1 = 3
- 5回目    i = 3+1 = 4
- 6回目    i = 4+1 = 5
- "        "

i <= 3 だからここまで

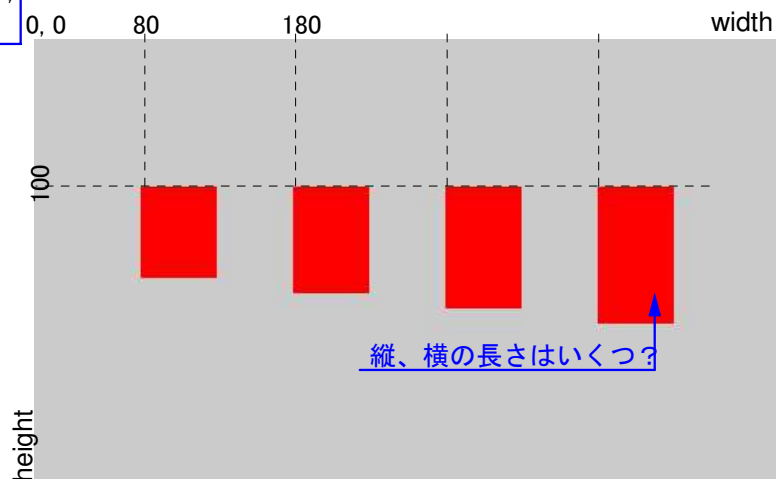
rect-4 で保存しよう。

### ミッション

上のコードを実行すると、図形が4つ描かれるね。for のなかではどんな計算式が動いているのかな？

- 1回目 rect(80+100\*0, 100, 50, (60+0\*10));
- 2回目            ?
- 3回目            ?
- 4回目            ?

と分けて、内容を紙に書いてみよう。そして計算して右の図とあっているか確認しよう。



### 3-ステップ 5 : 繰り返しを徹底的に理解する

ファイルから新規を開いて、X=100,Y=100 を基準点として  
あなたの名前を好きな色で、Y方向に5回繰り返して描いてみよう。

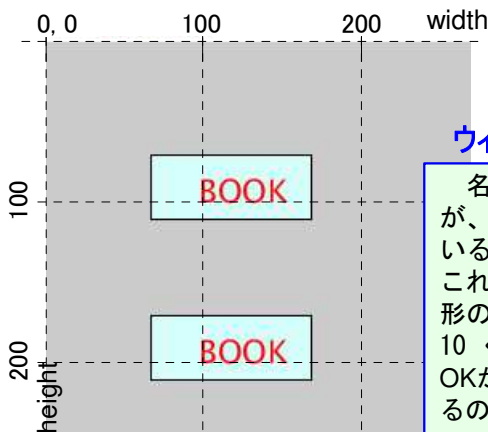
ヒント

1. size(700, 600); にしよう。
2. 繰り返しには for を使うんだね。
3. アルファベットは " " で囲まないと文字がでないぞ。
4. アルファベットの大きさは、textSize という命令を使うよ。
5. 文字の基準点はどこだ？
6. Y方向に繰り返すんだから、Y座標に変数を掛ければ良いんだ。
7. 右の図のBOOKのように、あなたの名前が並べば成功だ。

ミッション: 難問だよ

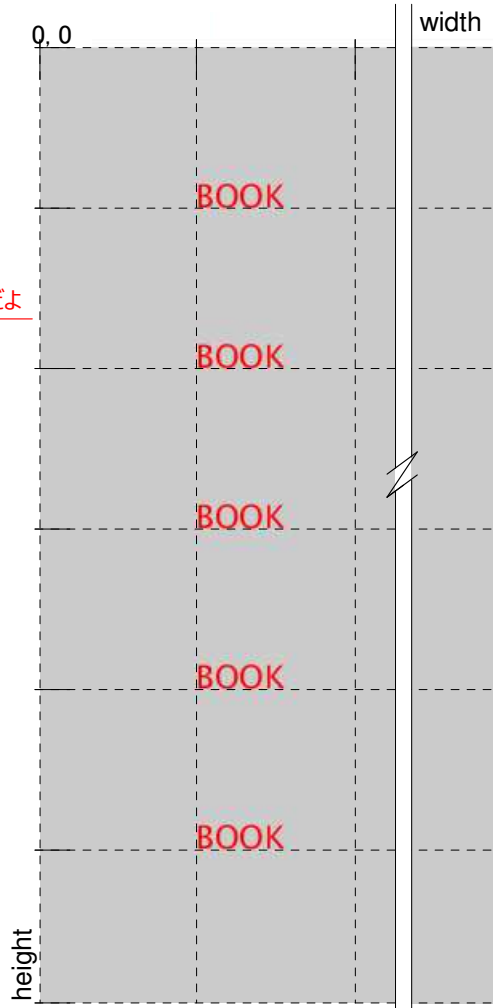
!! これだけ大文字だよ

1. あなたの名前を好きな色の四角形で囲ってみよう。
2. for 文から最後の行までの間に2行のコードを入れるんだよ。
3. 四角形の基準点はどこだ？
4. 名前より四角形の方が短かったら、横に長くすれば良いんだ。
5. Y方向に繰り返すんだから、Y座標に変数を掛けるんだよね。
6. できたら、my-name で保存しよう。



ウインドウの一部だ

名前と四角形の中心が、ちょっとズれているね。  
これを直すのは、四角形の基準点をY方向に10くらい移動すればOKかな。ここを調整するのが難題だね。



### 次回の予告

次回は、繰り返した図形に別々の色をつけてみよう。  
if という命令をつかうんだけど、ちょっと難しくなるよ。  
団子を食べる夢を見ながら、次回を楽しみしてね。

Bye-bye!

