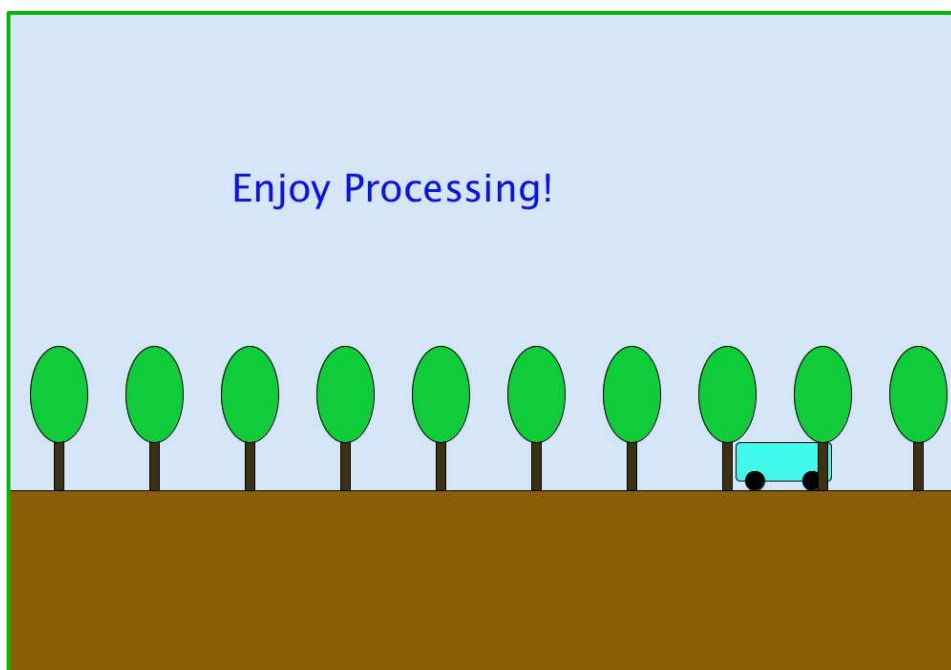


Processing

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);
```

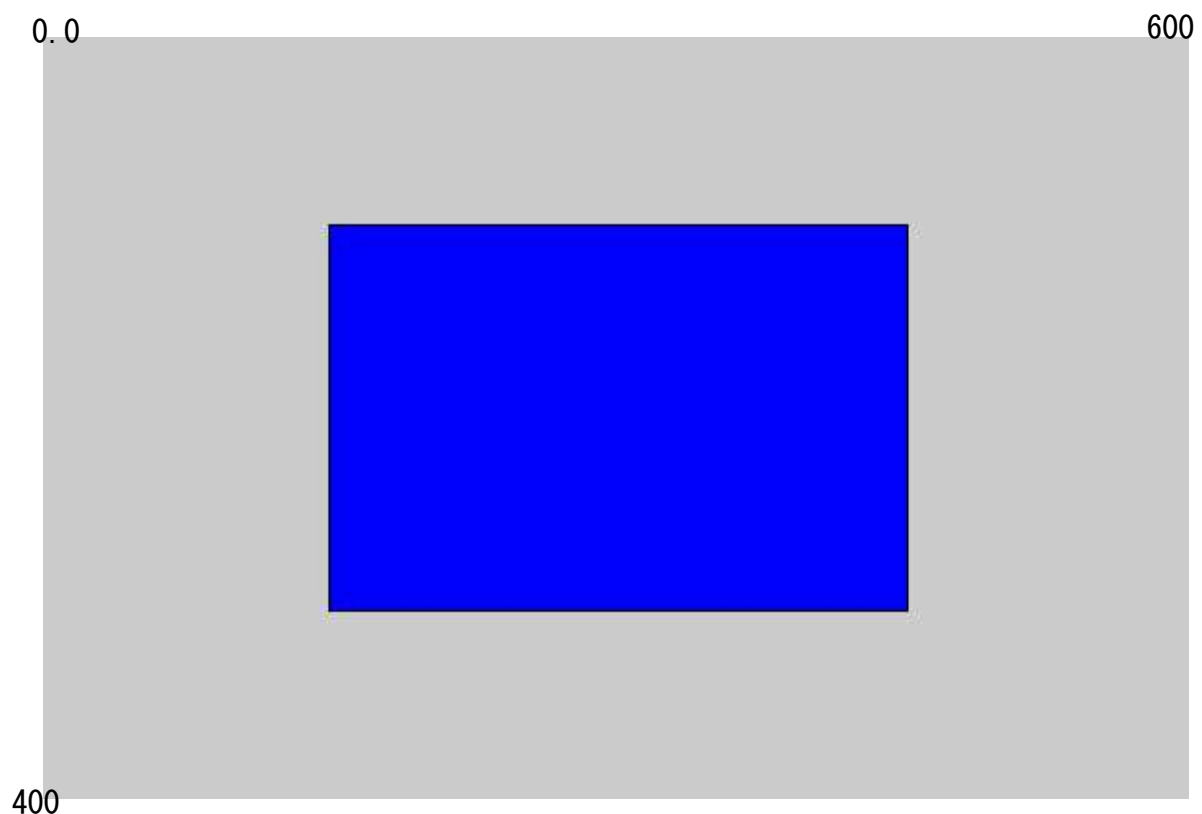
上のコードに下の 2 行目を加えて実行してみよう。

```
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。

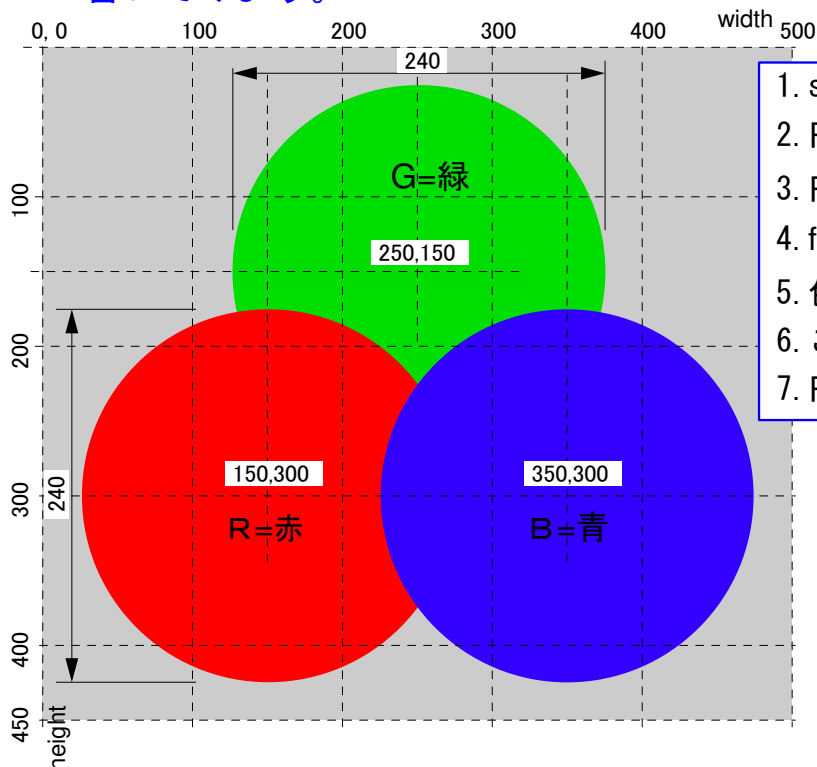
ミッション

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。



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、ヒントを見ながら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。

下のコードが参考になるかな。

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、く a は int (イント) つまり整数だと限定する。もし、小数なら float (フロート) を使うんだ。
int は integer (インテジャー: 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、どう変わるか実行してみよう。

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、何種類使っても良いんだ。

3 ステップ 2 : 変数を使いこなそう

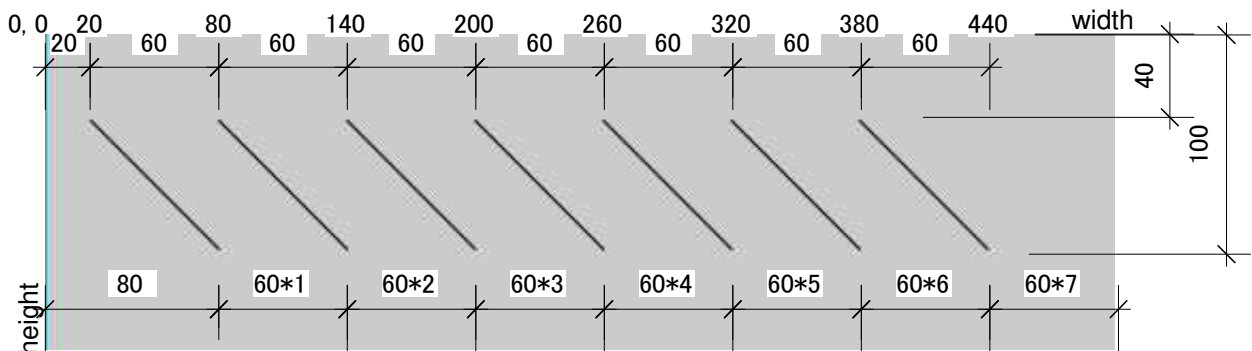
ファイルから新規を開いて、下図のコードを書いてみよう。

```
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);
```

この間は、3行目をコピーだよ。

変数で対応できるか

左のコードは、下のような図形になるね。
7本の直線だから、1本ずつコードを書いても良いけれど、1000本となると！
そこで、変数を使ってみよう。
左のコードに変数の宣言をしよう。



下のコードは書く必要ないよ。

```
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);
```

60 ずつ大きくなっている。

ただの変数じゃない!!

変数を使っても、コードの行数は変わらない。ただの変数をつかう場合じゃないね。

左のコードの
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。
こんな規則性ある場合は、次のように for
を使って、forの()の中で変数を使うんだ。

3-ステップ 3 : 繰り返しをもっと深く…for だよ

ファイルから新規を開いて、下のコードを書いて実行してみよう。

```

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判るよね。

このコードの意味は、下を読んでね。

```

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 になるまで繰り返す
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

```

もし、整数以下(小数)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して実行してみよう。

```

1 size(700, 700);
2
3 for(int a=1; a <= 6; a=a+1) {
4   ellipse(200, 100*a, 80, 80);
5 }

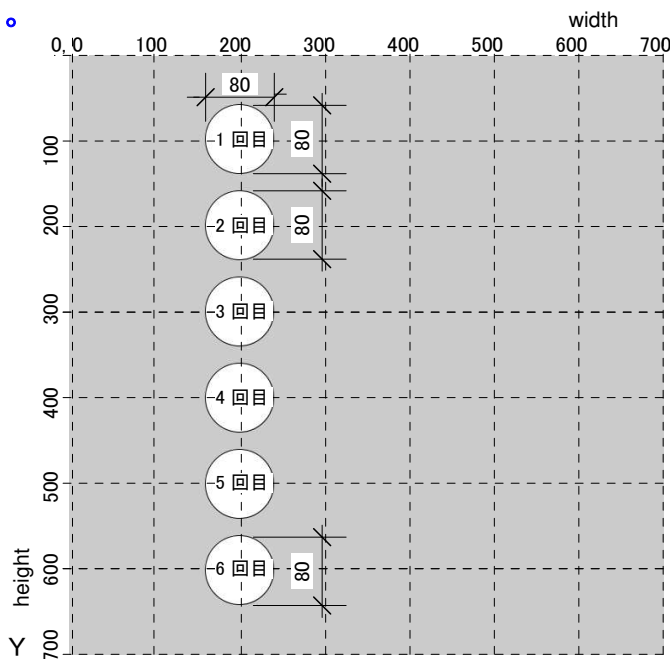
```

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

- 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。
- 円に好きな色を付けてみよう。
- for-1 と名前を付けて保存しよう。

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

println() は、print line を短くした形で、
コンソールに計算結果を出す命令だよ。

計算結果は下のコンソールにでるよ

1. 3. 7. 9. 10. は各自で確認してね。

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

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

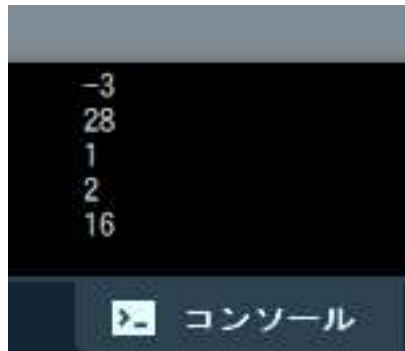
% は割った余りだけを返してくる

カッコがあるときはカッコの中から先に計算する

Lの小文字は l で、
iの大文字は I だよ。

保存は不要だよ。

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。



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、a < 5 は4までで、5は含まないんだ。
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。

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。

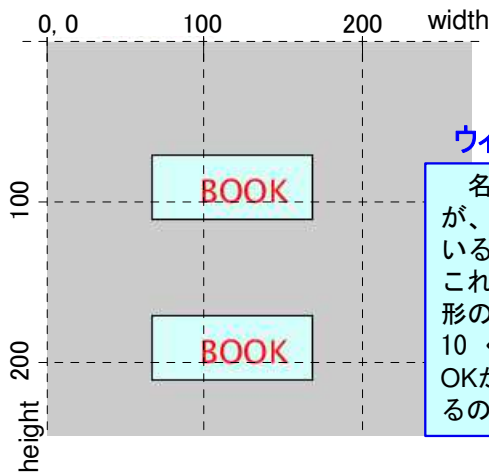
ヒント

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

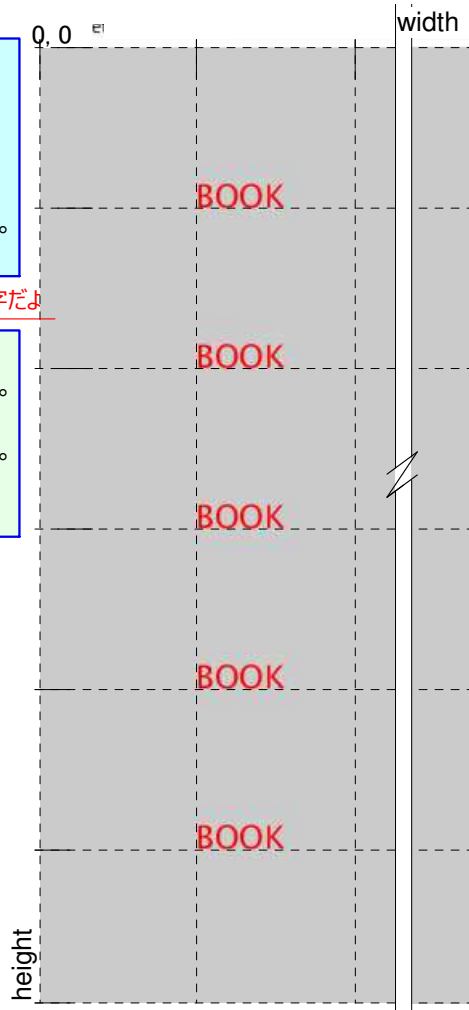
1. あなたの名前を好きな色の四角形で囲ってみよう。
2. for 文から最後の行までの間に2行のコードを入れるんだよ。
3. 四角形の基準点はどこだ？
4. 名前より四角形の方が短かったら、横に長くすれば良いんだ。
5. Y方向に繰り返すんだから、Y座標に変数を掛けるんだよね。
6. できたら、my-name で保存しよう。

!! これだけ大文字だよ



ウィンドウの一部だ

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!

