Processing

第３回

松田小学校／寄小学校

３ーステップ　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?

```
size(700, 700);
fill(0, 0, 255 );
rect(50, 50, 200, 400);
fill(255);
rect(250, 50, 200, 400);
fill(255, 0, 0);
rect(450, 50, 200, 400);
```

２行目と３行目をコピーして、
４行目と５行目、６行目と７行目に
ペーストして、数字を訂正するんだ。

ミッションー１

アンケート用紙の裏に左の国旗を描いてみよう。

France　で保存

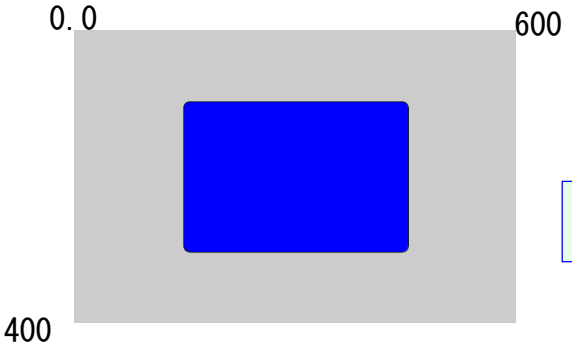
ファイルから新規を開き、次のコードを打ってみよう。

```
size(600, 400);
fill(0, 0, 255);
rect(150, 100, 300, 200, 10);
```

上のコードに下の２行目を加えて実行してみよう。

```
textSize(50);
text("XXXXXX", 200, 200);
```

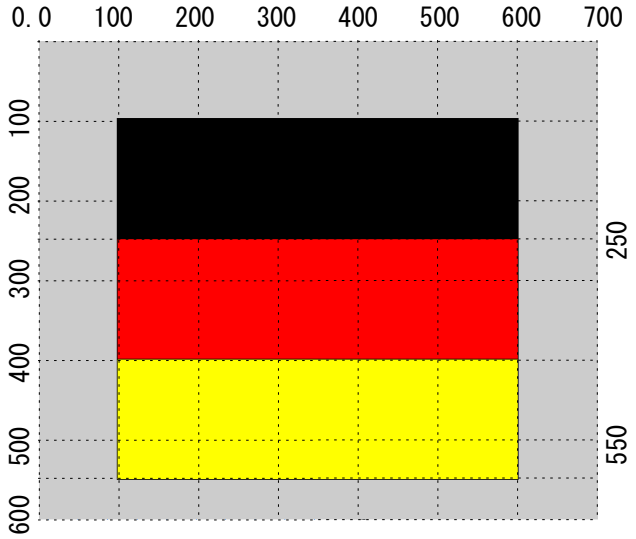
xxxxxx には、あなたの名前をローマ字で入れてね。



ミッションー２

４行目に何かコードを書いて、名前が見えるようにしよう。

name-1　で保存



ミッションー３　ファイル→新規

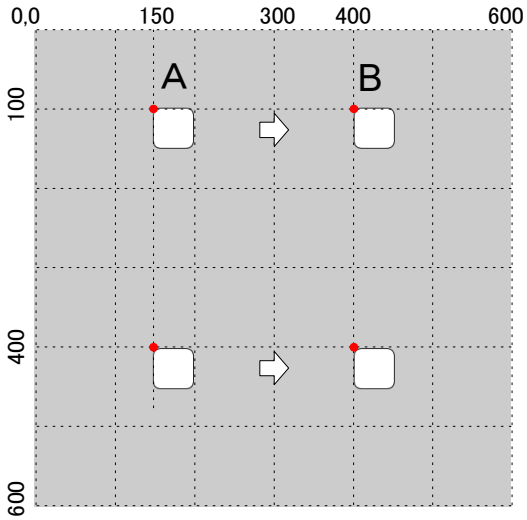
左の旗になるようにコードを書いてみよう。

Germany　で保存

３ーステップ　１：変数(=ｱﾙﾌｧﾍﾞｯﾄ)を使ってみよう

ファイルから新規を開いて、下のコードを打って実行してみよう。

```
size(600, 600);
rect(150, 100, 50, 50, 10);
rect(150, 400, 50, 50, 10);
```



ＡをＢへ移動するには、空欄には何を？

```
size(600, 600);
rect(□, 100, 50, 50, 10);
rect(□, 400, 50, 50, 10);
```

もし、四角が100コあったら！！

```
rect(100, 100, 50, 50);
rect(100, 200, 50, 50);
rect(100, 300, 50, 50);
rect(100, 400, 50, 50);
          〃
          〃
rect(100, 10000, 50, 50);
```

変数を使ってみよう。

最初に変数の宣言をすれば、プログラムの中で使うデータや値を保存するための変数(＝箱)を使えるんだ。変数にはアルファベットなら何でも何文字でもＯＫ！

宣言の形　int a;　とか　int baka;　でもOK。 int は integer の省略形で、整数という意味だ。インテジャー って読むよ

下のコードを打って実行してみよう。

```
int a=100;
size(600,600);
rect(a,100,50,50);
rect(a,200,50,50);
rect(a,300,50,50);
rect(a,400,50,50);
rect(a,500,50,50);
```

コピペして直しだ

X 座標を変数 a にして使う。

```
int a;     a=100;
```

分けて書いてもＯＫだ。

ミッション

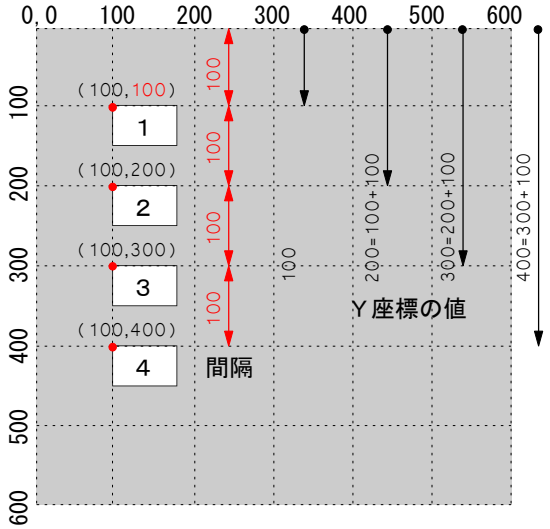
① int a = 100; を int a =300; や int a = 500; にして実行してみよう。
② 四角形の大きさを変数 b を使って、int b=50; で実行してみよう。b=80; では？

３ーステップ　２：変数の応用だ=繰り返し

```
int a;
    a=100;
size(600, 600);
rect(a, 100, 80, 50);
rect(a, 200, 80, 50);
rect(a, 300, 80, 50);
rect(a, 400, 80, 50);
```

変数 a を使う宣言

右図１のＸ座標



上のコードのＹ座標は、初めが 100 で 100 ずつ大きくなる。間隔が同じだね。間隔が同じという規則性がある場合は、繰り返しのコード for(～) を使うと簡単になるよ。

新規を開いて、下のコードを打って実行してみよう。コンソールにでたかな？

```
int i;
for(i=100; i<=400; i=i+100){
  println(i);
}
```

変数 i を使う宣言

for(～）の意味は、i を 100 から 400 まで、100 を加えながら繰り返せ。

変数 i の値をコンソール＝(黒い部分)に打ち出せ

for(～）の各意味は、下の通りだ。

for	繰り返しの開始記号
(i= 100;	変数 i を 100 (＝初期値)から
i <= 400;	変数 i が 400 以下なら { }内を実行する
i = i+100)	{ }内の i に 100 を足して更新し続ける
{println(i);}	{ }内の i の変化をコンソールに書く命令

ミッション

1．上のコードの、i = 100 を i = 200 に変えて実行するとどう変わる？
2．次に、i <= 600 に変えて実行すると、コンソールはどう変わる？
3．次に、i = i+200　と変えて実行すると、コンソールはどう変わる？
4．最初の値が 300 で、1100 以下まで、400 ずつ大きくなるように、下のコードの空欄を埋めてみよう。実行するとコンソールの数字はどうなったかな？

```
int i;
for(i=    ; i<=    ; i=i+    ){
  println(i);
}
```

（初期値）　（繰り返す上限）　（更新式）

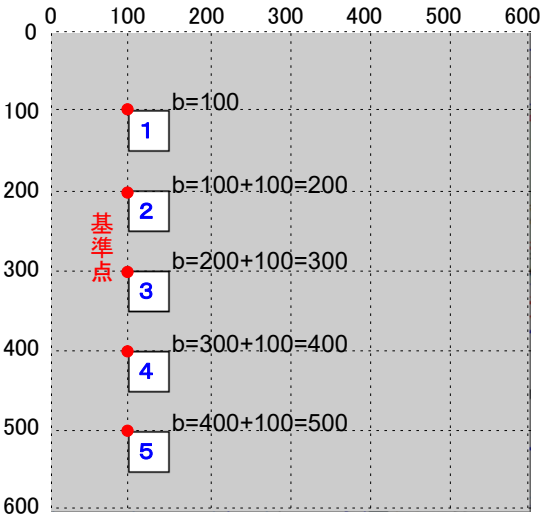
３ーステップ　３：繰り返しの式の意味…for だよ

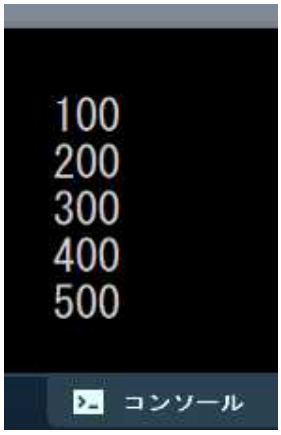
新規を開いて、四角形を描くコードと for(　) を合体せてみよう。

```
1 int a=100;
2 int i;
3 size(600, 600);
4 for(i=100; i<=500; i=i+100){
5   rect(a, i, 50, 50);
6   println(i);
7 }
```

四角形のＸ座標に変数 a を使う宣言

四角形のＹ座標に変数 i を使う宣言





b = 100 のとき、b = b + 100 は

１回目　b = 100 代入
２回目　b = 100+100 = 200
３回目　b = 200+100 = 300
４回目　b = 300+100 = 400
５回目　b = 400+100 = 500

b <= 500; だからここまで
もし、b<500; なら、
500を含まないから、４回目まで。

プロセッシングで使う比較記号

＝　等しい(等号)ではなく、代入記号だよ
==　等しい(等号)
!=　等しくない
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で、右辺の数字や記号を左辺に代入するのに使うんだよ。

< や > の後ろに = が付くと、付かないでは意味が違うから気をつけよう。
たとえば、a<5 は５未満で、５は含まないけど、a<=5 は５以下で、５を含むんだ。

for(～) の数字を変えれば、いくら図形が増えても対応できる

３－ステップ ４：繰り返しをもっと深く…for だよ

前のコードを手直しして実行しよう。変数宣言は for( ) のカッコの中に入れてもＯＫだ。

```
int a=80;
size(800, 200);
for(int i=100; i<=700; i=i+100) {
  circle(i, 100, a);
}
```

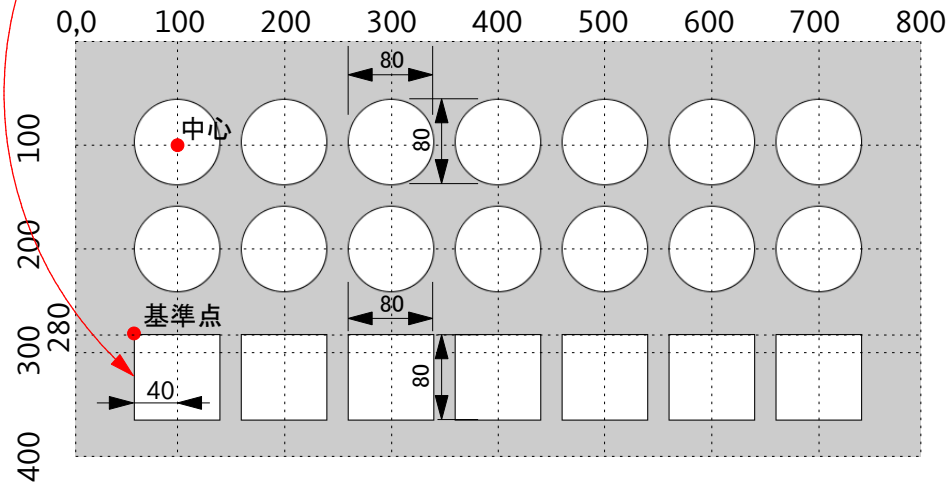
上のコードの意味は、右を読んでね。

```
for                  //繰り返しの開始記号
(int i=100;          //変数 i を 100(＝初期値)から
 i <= 700;    //変数 i が 700 になるまで { } 内を実行する
 i = i+100)          // i を 100 ずつ大きくして繰り返す
{circle( i, 100, a);}     //変数 i を変化させながら描く
```

for( ) の { } の中には色々なコードを入れることができるんだ。加えてみよう。

```
int a=80;
size(800, 400);
for(int i=100; i<=700; i=i+100) {
  circle(i, 100, a);
  circle(i, □, a);
  rect(i-40, 280, a, a);
}
```

ここも直したよ。



ミッション

１．size(800, 800);で、最初の円の座標が X=100, Y=100で始まる直径 80 の円を Ｙ座標に平行に７ヶ描いてみよう。

２．size(800, 800);で、最初の円の座標が X=100, Y=100で始まる直径 80 の円を 斜め下 45° に向けて７ヶ描いてみよう。

３．for-1 と名前を付けて保存しよう。

３ーステップ　５：頭の体操をしてみよう

下のプログラミングをした時に、計算結果にどんな数字が入るか考えてみよう。

a=100;
代入すると
a はいくつ？

a=1;
代入すると
100*a はいくつ？

＊は掛ける（×と同じ）
／は割る（÷と同じ）

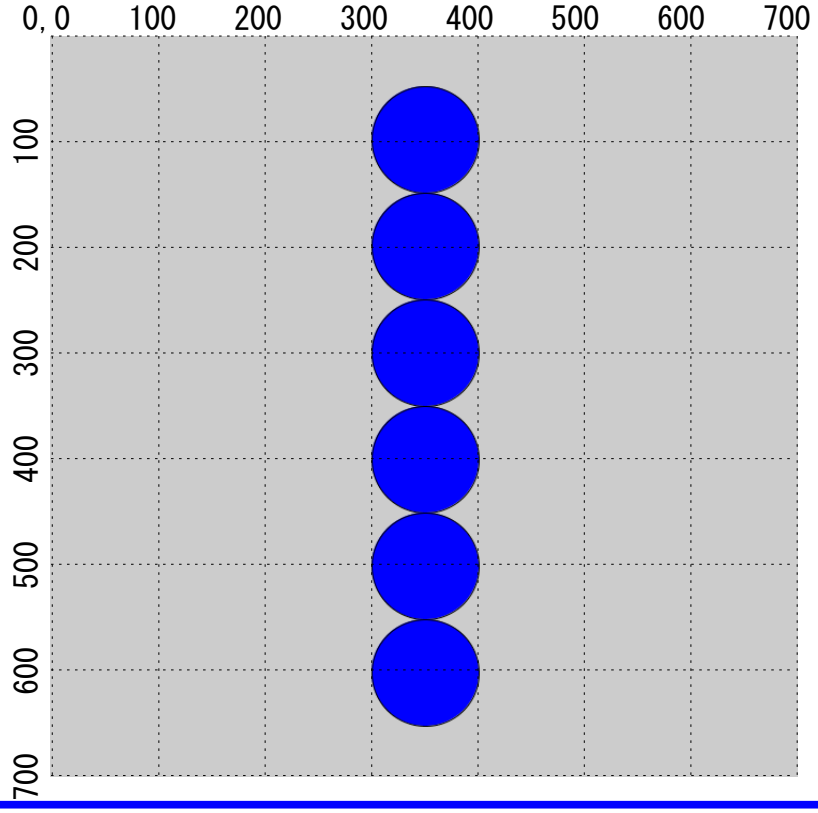
ファイルから新規で、下のコードを打ってみよう。

```
size(600, 200);

for(int a=100; a<=500; a=a+100){
  circle(a, 100, 50);
}
```

上のコードを手直しして、同じ結果になるように空欄を埋めてみよう。

```
size(600, 200);

for(int a=1; a<=5; a=a+1){
  circle(    *a, 100, 50);
}
```

下の図になるように、上のコードを手直ししてみよう。



balls-6 で保存しよう。

３ーステップ　６：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 、下のコードを打ってみよう。

```
size(700, 700);
for(int a=100; a<=600; a=a+100) {
  for(int b=100; b<=600; b=b+100) {

    circle(a, b, 80);

  }}
```

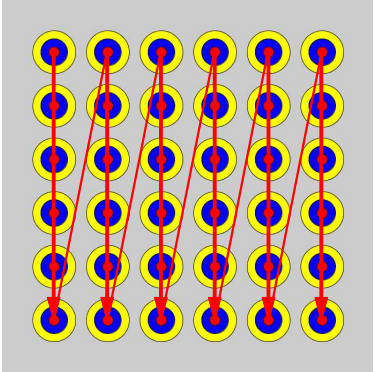
繰り返しのなかに、繰り返しコードもOKだ。

ミッション

1. ４行目に、fill(255,255,0); と打って、実行してみよう。
2. ６行目に、circle(a,b,50); と打って、実行してみよう。
3. 内側の円だけに、好きな色を付けてみよう。
4. 内側の円の中に、直径20の円を描いて好きな色を付けてみよう。
5. できたら、**circle** で保存しよう。

発展学習ー繰り返しの順番

同心円が３６コ描けたけど、どういう順番で描いているのかな。
それを確かめるには、１０行目(コードの最後の行)に
`println(a, b);` と打ってみれば分かるよ。
コンソールには左側＝ａ、右側＝ｂの数字が出ているんだ。
ａが100で６回描いている間に、ｂは100～600まで描いて、次に
ａは200で６回描いて、ｂは100～600まで順に描いているんだよ。



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、次回を楽しみしていてね。

Bye-bye!