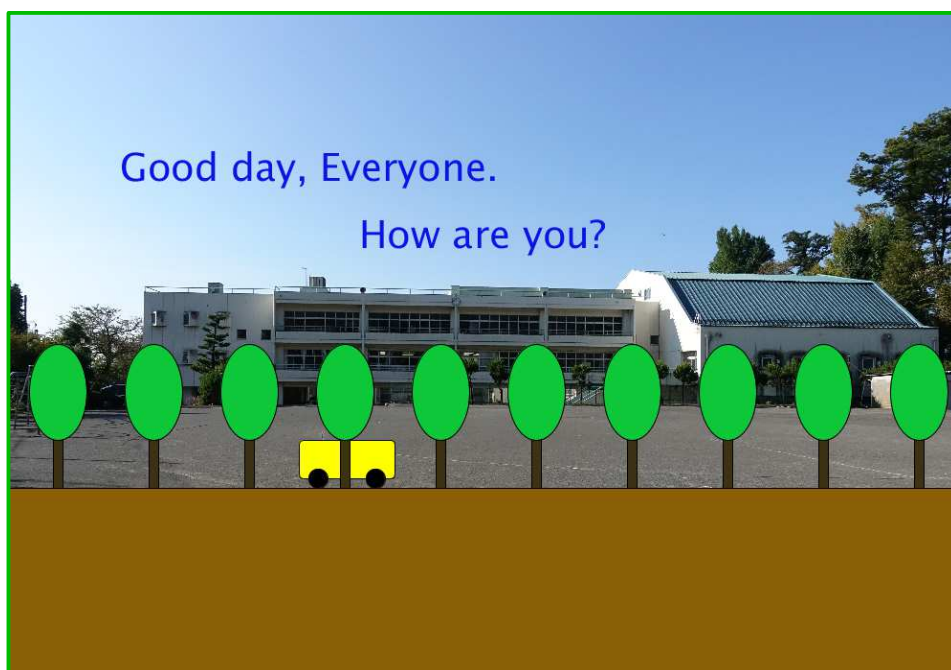


Processing

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, 10);
```

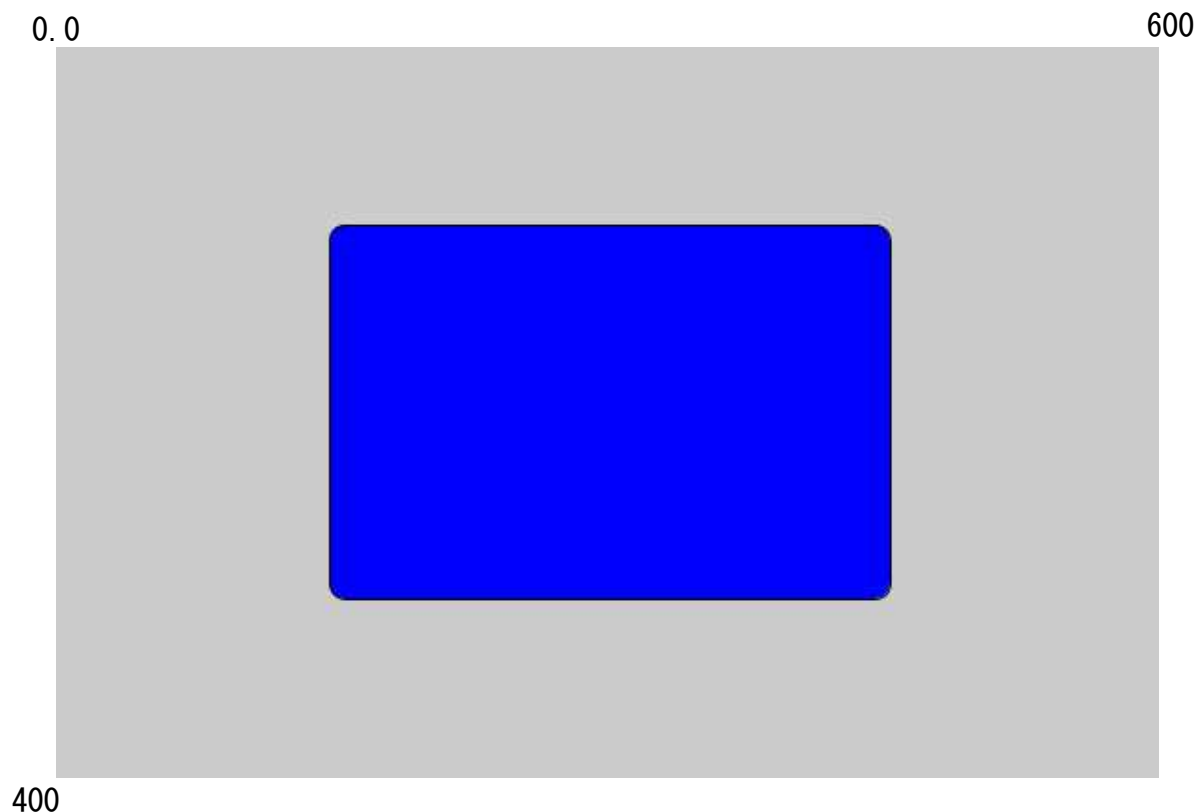
上のコードに下の2行目を加えて実行してみよう。

```
5 textSize(50);  
6 text("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

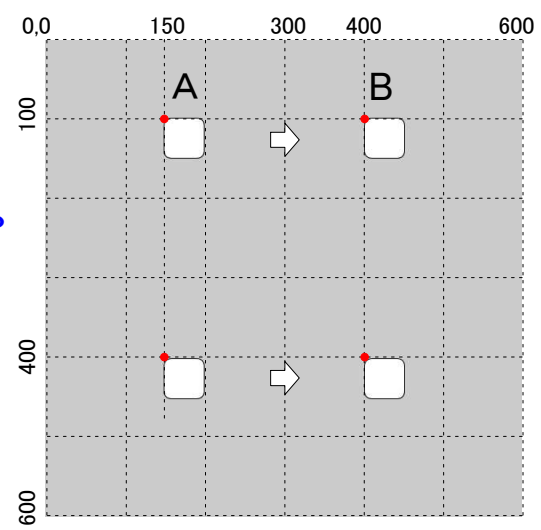
あれ！あなたの名前が見えないね。どうしたら見えるようになるだろうか。
4行目に何かコードを加えて、見えるようにしてみよう。



3ステップ 1 : 変数(=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、下のコードを打って実行してみよう。

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



AをBへ移動するには、空欄には何を？

```
1 size(600, 600);
2 rect( , 100, 50, 50, 10);
3 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; または int a=100;

下のコードを打って実行してみよう。

```
1 int a=100;
2 size(600, 600);
3 rect(a, 100, 50, 50);
4 rect(a, 200, 50, 50);
5 rect(a, 300, 50, 50);
6 rect(a, 400, 50, 50);
7 rect(a, 500, 50, 50);
```

X 座標を変数 a にして使う。

```
1 int a; a=100;
分けて書いてもOKだ。
```

ミッション

- ① a = 100; を a = 300; や 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、3分間で下図のコードを書いてみよう。

```
1 int a=80;
2 size(800, 200);
3 circle(100, 100, a);
4 circle(200, 100, a);
5 circle(300, 100, a);
6 circle(400, 100, a);
7 circle(500, 100, a);
8 circle(600, 100, a);
9 circle(700, 100, a);
```

コピーして直した

いくつ書けるかな！
みんなで競争してみよう。

ただの変数じゃない!!

左のコードの
X座標は、100 ずつ大きくなる。
こんな規則性ある場合は、
次のように for() を使うんだ。

100 ずつ大きくなっている。

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 int a=80;
2 int i;
3 size(800, 200);
4 for (i=100; i<=700; i=i+100) {
5     circle(i, 100, a);
6 }
```

circle の直径に変数 a を使う宣言

circle の中心の X 座標に変数 i を使う宣言

後の { } の i に 100 を足し続ける。

(更新式)

最初の i は 100 から

(初期値)

i が 700 になるまで { } 内を実行

(繰り返す上限)

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。

i = 100 のとき、i = i + 100 は

1回目	i = 100 代入
2回目	i = 100+100 = 200
3回目	i = 200+100 = 300
4回目	i = 300+100 = 400
5回目	i = 400+100 = 500
6回目	i = 500+100 = 600
7回目	i = 600+100 = 700

i <= 700; だからここまで

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。
< 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。

3-ステップ 3 : 繰り返しをもっと深く…for だよ

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

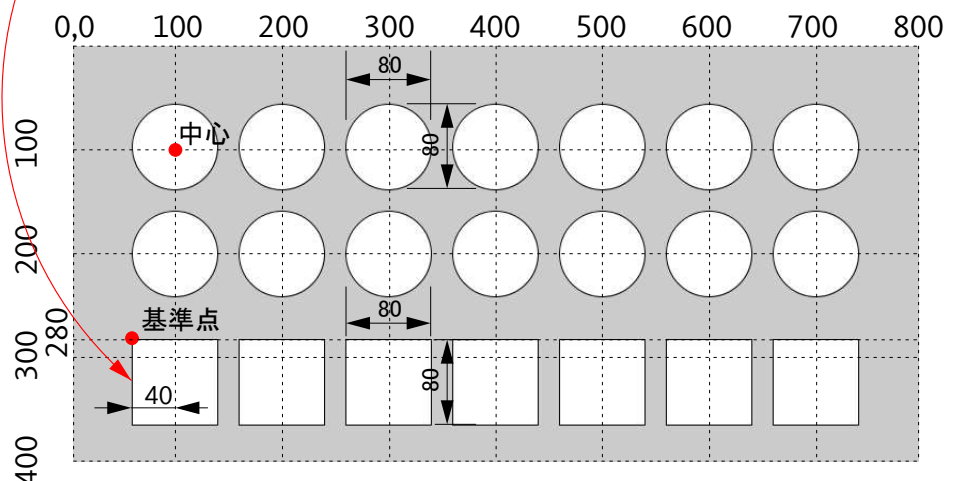
```
1 int a=80;
2 size(800, 200);
3 for(int i=100; i<=700; i=i+100) {
4     circle(i, 100, a);
5 }
```

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)   //{ } 内の i を 100 ずつ大きくして繰り返す
{circle(i, 100, a);} //{ } 内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。

```
1 int a=80;
2 size(800, 400); ←ここも直したよ。
3 for(int i=100; i<=700; i=i+100) {
4     circle(i, 100, a);
5     circle(i, , a);
6     rect(i-40, 280, a, a);
7 }
```



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 個描いてみよう。
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 個描いてみよう。
3. `for-1` と名前を付けて保存しよう。

3 ステップ 4 : 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; a=1;
↓ 代入する ↓ 代入する
a= ? 100*a= ?

← 同じだよ →

* は掛ける (×と同じ)
/ は割る (÷と同じ)

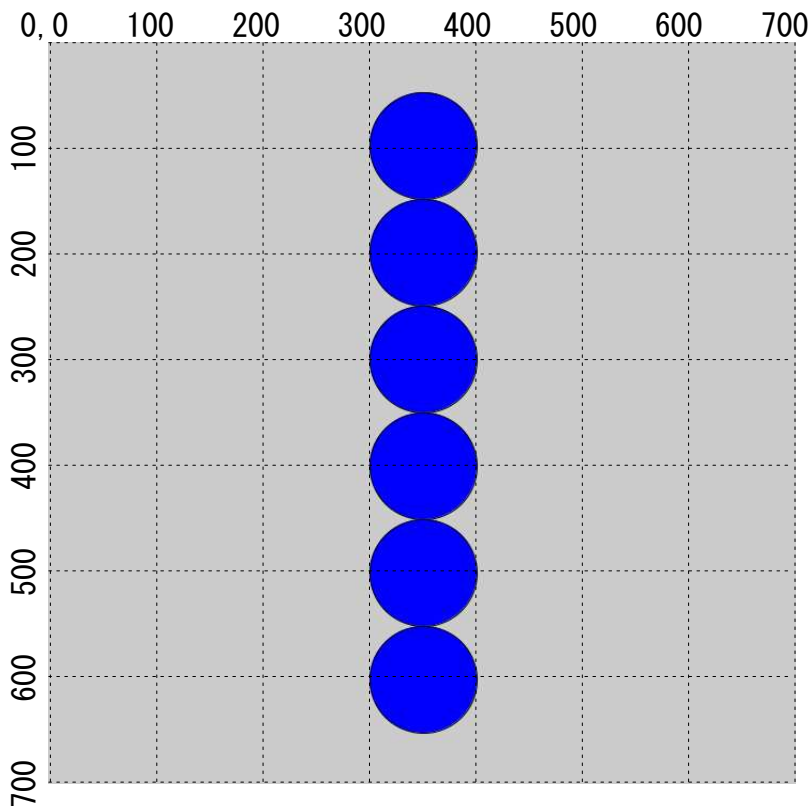
ファイルから新規で、下のコードを打ってみよう。

```
1 size(600, 200);  
2  
3 for(int a=100; a<=500; a=a+100) {  
4   circle(a, 100, 50);  
5 }
```

上のコードを手直して、同じ結果になるように空欄を埋めてみよう。

```
1 size(600, 200);  
2  
3 for(int a=1; a<=5; a=a+1) {  
4   circle(*a, 100, 50);  
5 }
```

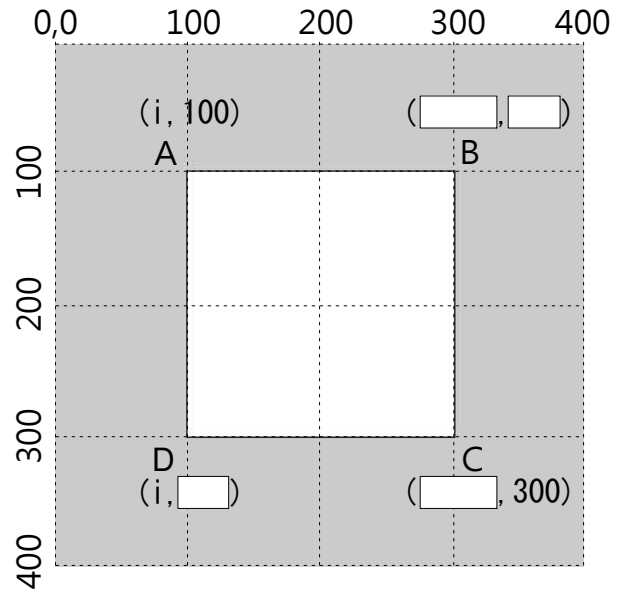
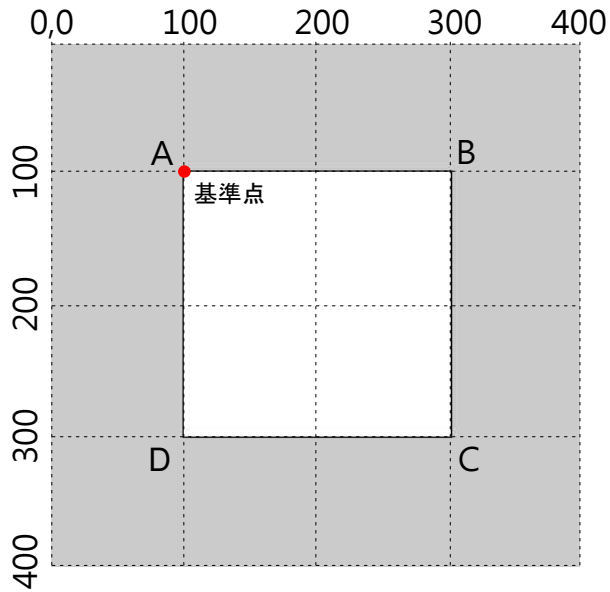
下の図になるように、上のコードを手直してみよう。



balls-6 で保存しよう。

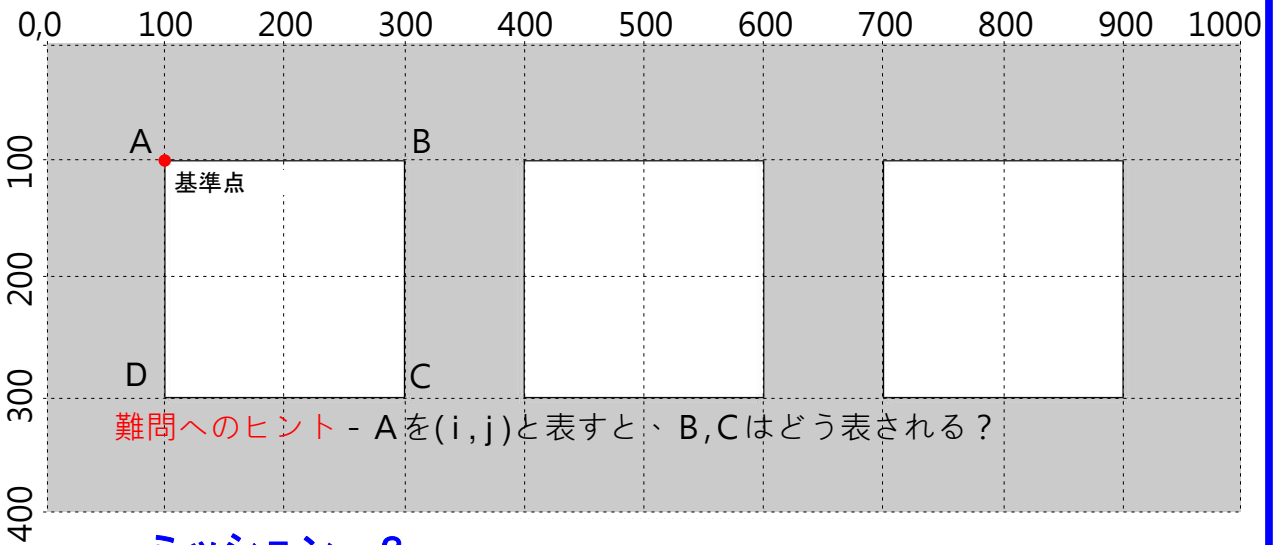
3ステップ 5 : 四角形を繰り返してみよう

ファイルから新規を開いて、左下の正方形のコードを書いてみよう。



ミッション-1

1. 右図の空欄を埋めよう。そして、変数 i を使ったコードに書き換えてみよう。
2. 下図のようになるコードを書いてみよう。



ミッション-2

1. 上の四角形に好きな色を付けてみよう。
2. 四角形をY座標にそって、縦に並べてみよう。
3. **難問!** 下のコードを埋めて、縦横3ケずつ計9ケの四角形を描いてみよう。

```

1 size(1000, 1000);
2 for(int i=1; i<=7; i=i+3) {
3   for(int j=1; j<=7; j=j+3) {
4     rect( , , 200, 200);
5   }

```

for () には図だけではなく
for () を入れることもできる

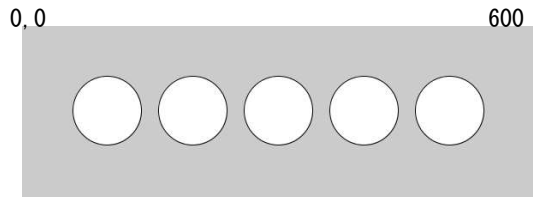
3-ステップ 6 : 繰り返しの繰り返し

新規のファイルを開いて、下のコードを打ってみよう。

下のコードは、右のように円が5つ繰り返されるね。

```

1 size(600, 600);
2
3 for(int x=100; x<=500; x=x+100) {
4   circle(x, 100, 80);
5 }
    
```



for()に続く{ }のなかには、図形のコードだけではなく、式のコードも入れることができるんだ。それが下のコードだよ。

下のコードの3行目のfor文は、y を変数としているけど、circle のなかには y が使われていないね。これでは実行結果は上の図と同じだ。円のY座標に変数 y を入れると、繰り返しが繰り返されて下の図のように変わるんだ。

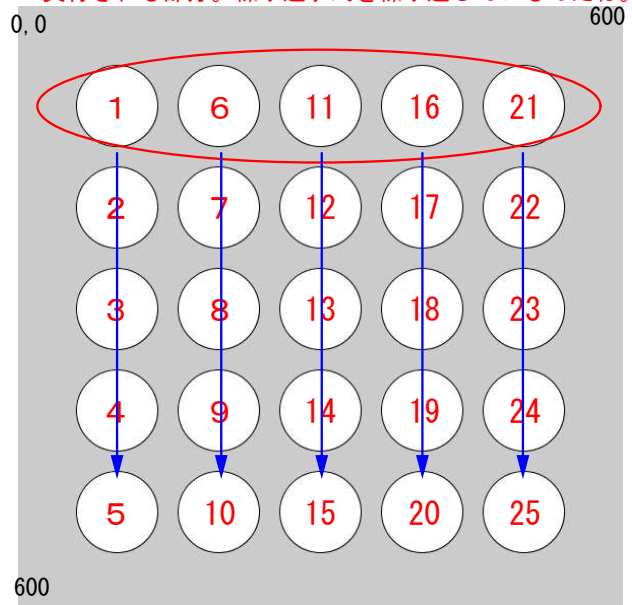
```

1 size(600, 600);
2 for(int x=100; x<=500; x=x+100) {
3   for(int y=100; y<=500; y=y+100) {
4     circle(x, 100, 80);
5   }
6 }
    
```

このY座標 100 を変数 y に変えて実行すると、縦にも円が5つ繰り返されて、下のようになるよ。

2行目の for(int x=100; x<=500; x=x+100) で実行される部分。繰り返す式を繰り返しているんだね。

4行目の円のY座標に変数 y をいれると、まず2行目の for文が実行されて、1の円が描かれる。次に3行目の for文が実行されて、2~5までの円が描かれ終わって、6に戻ってくるんだ。次に同じように7~10の円が描かれ続けるというわけだ。そして、次々に縦が繰り返されていくんだ。



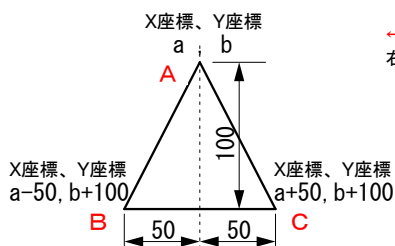
3行目の for(int y=100; y<=500; y=y+100) で実行される部分。縦に1回だけ繰り返して上に戻っているね。

ミッション

- 3行目と4行目のあいだに、fill(); を入れて好きな色にしてみよう。
- 70×70の四角形を、縦横ともに100間隔で、横に8ヶ、縦に6ヶ並べてみよう。size が変わるかな？
- size を full screen にして、底辺が100、高さが100の二等辺三角形を、ウィンドウ全体に描いてみよう。

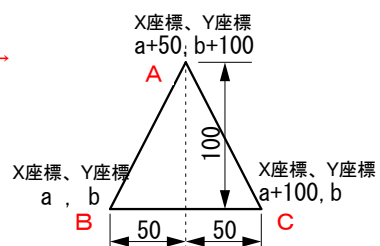
③へのヒント

- fullScreen にしたときは、x<=width; y<=height; にすると、横幅と縦いっぱいひろがるよ。
- 三角形の頂点の座標は、下のよう考えるんだ。



←左はAを基準に考えたときで、右はBを基準に考えたときだよ。→
結果は同じになるよ。

Cを基準にするとどうなるかな？



3ステップ 7 : 繰り返しを徹底的に理解する

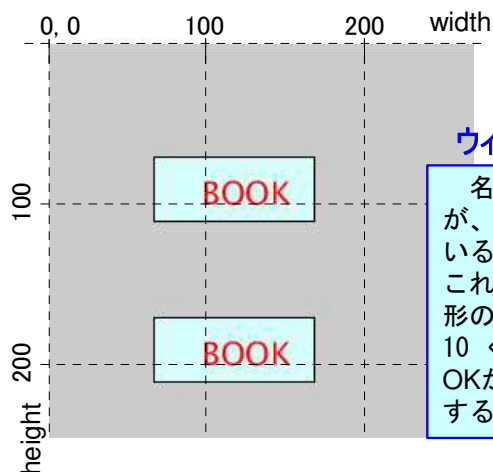
ファイルから新規を開いて、X=100,Y=100 を基準点として
あなたの名前を好きな色で、Y方向に5回繰り返して描いてみよう。

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。 !! これだけ大文字だよ
3. アルファベットの大きさは、textSize(); 命令を使うよ。
4. アルファベットは " " で囲まないと文字がでないぞ。
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座標に変数を使えば良いんだ。
7. 右の図のBOOKのように、あなたの名前が並べば成功だ。

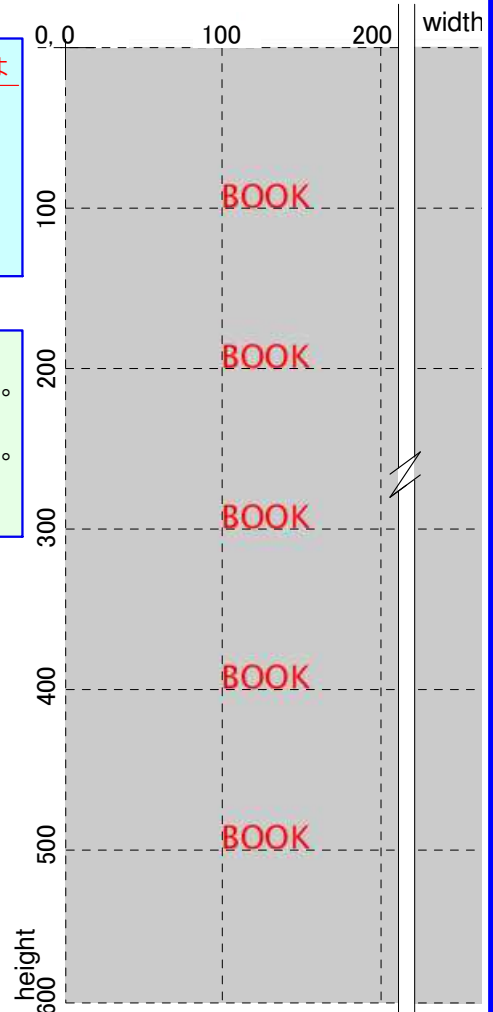
ミッション: 難問だよ

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座標に変数を使うんだよね。
6. できたら、my-name で保存しよう。



ウインドウの一部だ

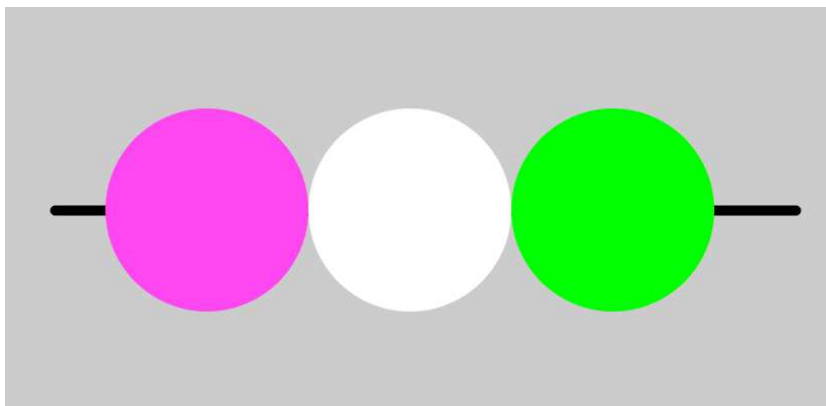
名前と四角形の中心が、ちょっとズれているね。
これを直すのは、四角形の基準点をY方向に10くらい移動すればOKかな。ここを調整するのが厄介だね。



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、次回を楽しみしていてね。

Bye-bye!



第3回 変数と繰り返し

① 何でも入れることができるから、変数はよく箱といわれます。しかし、箱ではイメージが具体的に過ぎ、本ガイドではあえて箱という表現は使っていません。むしろ、繰り返しなどから変数を使うに至る必然性を導きだすように努めています。

初心者のうちは、変数に意味のある単語を使うと意味にとらわれて、他のコマンドとの区別がつかなくなると考え、アルファベット1文字を使うようにしています。慣れてきたら、内容を反映した単語を使って下さい。

② 繰り返し(for 文)の初期値は明確ですが、変化の範囲(上限)が明確にならない嫌いがあります。最後の値は、上限をでても成り立つようです。とくに範囲指定の数字が3桁になっている場合には、収まらないかどうか注意が必要です。

③ `size(800,800); for(int a=1; a <= 6; a = a+1) {ellipse(200,100*a,80,80);}` と
`size(800,800); for(int a=100; a <= 600; a = a+100) {ellipse(200,a,80,80);}` とは同じ結果になります。for 文の中で1だったものを、円を描く方で100倍しただけですから結果は同じです。

④ `for(int a=1; a <= 6; a = a+1) {ellipse(200,100*a,80,80);}` という具合に1行で書くのと `for(int a=100; a <= 600; a = a+100) {
 ellipse(200,a,80,80);
}` と数行に分けるのはまったく同じです。{以降を改行するのが、業界の習慣らしいので本ガイドでも改行していますが、1行で書いたほうが理解しやすいこともあります。なお、数行に分ける場合は、2行目から行頭を下げる習慣です。下げるのに全角の空白を使うとエラーになります

⑤ `a=a+1` は `a++`(インクリメント)、`a=a-1` は `a--`(デクリメント)でも表せますが、代入式の方が素直に理解できると考えて、インクリメントとデクリメントはあえて使用していません。

⑥ 計算記号や比較記号は、数学で使うものを基礎にしていますが、若干の違いがあります。とくに `=` (代入)と`==` (等号)の違いは強調する必要があります。`a = a+1` は右辺と左辺が違いながら数学の等号で結んでいます。数学ではこんな等式は成り立ちません。

プログラミングでは、`a=1;` は右辺を左辺に代入する意味です。初期値を `a=1;` とす

ると、1 を $a=a+1$ の右辺に代入して、 $a=1+1$ で $a=2$ になります。次に $a=2$ となった 2 を代入するので $a=2+1$ で $a=3$ になります。以降、これの繰り返しになっていきます。

```
⑦ size(800,800);
for (int i = 2; i <=8; i = i+1 ) {
    for (int j = 2; j <=8; j = j+1 ) {
        ellipse(80*i, 80*j, 50, 50);    } }

```

と言うように、for 文のなかに再度 for 文を入れることができます。入れ子と言います。

```
⑧ size(800,800); for (int
i = 2; i <=8; i = i+1 ) {
ellipse(80*i, 80*j, 50, 50); }

```

は、

$i = 2$ から $i \leq 8$ を判断し、正しい場合には $i = i+1$ をとばして、`{ellipse();}` にいって円を描きます。つぎに戻って $i = i+1$ を実行した後、また円を描き $i = i+1$ の実行を繰り返し、 $i \leq 8$ を満たさなくなると for 文の実行を止める順序でうごきます。

⑨ プロセッシングでは繰り返し、for 文の他に while 文、switch 文があります。しかし、for 文だけで十分にコードが書けますので、後二者は省略しています。

ステップ 0

```
1 size(600,400);
2 fill(0,0,255);
3 rect(150,100,300,200,10);
4 textSize(50);
5 fill(255);
6 text("Matsuda",200,200);

```

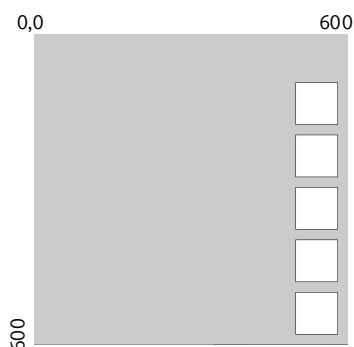
2行目の `fill(0,0,255);` が、文字にも効いているので同色になって目立たない。5行目に別の色を加えた。

← 加えた

ステップ 1 ミッション②

```
1 int a=500;
2 int b=80;
3
4 size(600,600);
5 rect(a,100,b,b);
6 rect(a,200,b,b);
7 rect(a,300,b,b);
8 rect(a,400,b,b);
9 rect(a,500,b,b);

```



ステップ 3 ミッション1 下のどれでも良い。

```
1 size(800,800);
2 for(int a=100; a<=700; a=a+100){
3     ellipse(100,a,80,80);
4 }
1 size(800,800);
2 for(int a=1; a<=7; a=a+1){
3     ellipse(100,100*a,80,80);
4 }
1 int b=80;
2 size(800,800);
3 for(int a=1; a<=7; a=a+1){
4     ellipse(100,100*a,b,b);
5 }

```

ステップ 3 ミッション1-2

```
1 int b=80;
2 size(800,800);
3 for(int a=1; a<=7; a=a+1){
4     ellipse(100*a, 100*a, b, b);
5 }
```

ステップ 5 ミッション1-2

```
1 int a=200;
2 size(1000,400);
3 for(int i=1; i<=7; i=i+3){
4     rect(100*i, 100, a, a);
5 }
```

ステップ 5 ミッション2-1

```
1 int a=200;
2 size(1000,400);
3 fill(0,255,0);
4 for(int i=1; i<=7; i=i+3){
5     rect(100*i, 100, a, a);
6 }
```

for文の前に色指定しても、
後で色指定しても、結果は
同じだが、実行される順番が違う。

```
1 int a=200;
2 size(1000,400);
3 for(int i=1; i<=7; i=i+3){
4     fill(0,255,0);
5     rect(100*i, 100, a, a);
6 }
```

ステップ 5 ミッション2-2

```
1 int a=200;
2 size(1000,1000);
3 fill(0,255,0);
4 for(int i=1; i<=7; i=i+3){
5     rect(100, 100*i, a, a);
6 }
```

ステップ 5 ミッション2-3

```
1 size(700,600);
2 for(int i=1; i<=5; i=i+1){
3     textSize(50);
4     fill(255,0,0);
5     text("Matsuda", 100, 100*i);
6 }
```

ステップ 6

```
1 size(700,600);
2 for(int i=1; i<=5; i=i+1){
3
4     fill(64,220,220);
5     rect(80, 100*i-48, 250, 60);
6
7     fill(255,0,0);
8     textSize(50);
9     text("Matsuda", 100, 100*i);
10 }
```

Y座標の位置を-48で
調節しているが、正確な
位置は実行してみないと
分からない。

ステップ3:下の for文
の四角形 i-40 を参照