

Processing で「変数」と「繰り返し」
のイメージをつかむ

課題 2011/4/18

- 1・下がりに円を5個描け。
- 2・右下がりには5個、左下がりに5個描け
- 3・美しいアート作品を作れ。

変数(数値を入れる箱)

宣言 //こういう名前の変数使うよ!

変数への数値の代入(書き込み)

= は、<=の意味だよ。

```
int a= 5; //(整数しか入れられない)変数aを宣言  
し a に初期値5を代入(書き込む)
```

```
int a; //(整数しか入れられない)変数aを宣言  
a=5; //a に5を代入
```

演算

```
a= a+7; //変数a の中身に7を足してaに書き込む
```

繰り返しfor

5回繰り返す


```
int a =0;
for (int i=0; i<5; i++) {
    a=a+ 3;
}
```

forの{ }内を5回くり返す

アート用プログラミング言語 (processing) よるプログラミング入門

準備 : step1:processingダウンロード、解凍

<http://processing.org/download/index.html> から
windows版をダウンロードし解凍。

フォルダー内の  クリックすると、
step1:ウインドウ(右図の右部)が現れ、
step2:プログラムを書き込み、
step3:三角マークでプログラム実行。
step4:図左のウインドウ
(左上隅が原点0,0)が現れ実行結果の表示。



課題：運まかせのデジタルアート制作

(作戦)

(1) 線、三角、四角、丸、を沢山重ね描き
(繰り返し)

(2) 場所、大きさ、色、透明度、のバリエーション
(乱数)

(プログラムのには)

(1) 繰り返しのfor

(2) 繰り返しの中で規則的に場所や色を変えるための、変数int

(3) 乱数random(...)

(4) (場合によっては)条件分岐のif

プログラム1:(線を引く)

- `line(10,10,80,80); //1行のプログラム`
- ここで、`//`マークは、コンピュータがその行の`//`以降を無視することを示す。
- `line(線開始x座標、線開始y座標、線終了x座標、線終了y座標);`

I. バリエーションの作り方

- プログラム1: (線を引く):

```
line(10,10,80,80); //座標(10,10)から(80,80)へ
```

- プログラム2: (変数を使う)

```
int st=10, en=80; // intは整数という意味  
line(st, st, en, en);
```

- プログラム3: (乱数を使う)

```
line(random(10),random(10),random(80),random(80));
```


変数

```
int st=10, en=80;    //プログラムで使う変数の宣言  
line(st, st, en, en); //プログラム本体
```

- 1行目の、「int st=10, en=10;」は stという整数を入れる変数(数値箱)と、enという整数を入れる変数を用意し、それぞれに初期値として、stに10を入れ、enに80を入れることを示す。
- (st=10; の=は等しいという意味ではなく、左矢印(<=)を簡単化したもので、書込み<代入>を意味する)

繰り返し

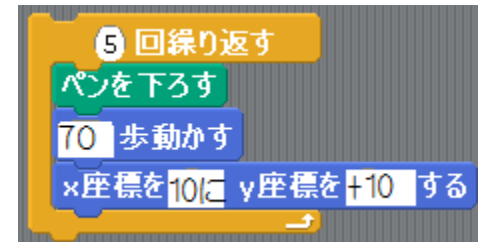
- プログラム3: 5本水平線を引く

```
line(10, 0, 80, 0);  
line(10, 10, 80, 10);  
line(10, 20, 80, 20);  
line(10, 30, 80, 30);  
line(10, 40, 80, 40);
```



- プログラム4: プログラムをfor文で書く

```
int y= 0;  
for(int i=0; i<5; i++){ //i++は、iに1を加えること  
    line(10, y, 80, y);  
    y= y+ 10; // yに10を加える(yは0,10,20,30,40と変わる)  
}
```



繰り返しfor

10個図形（線、円など）を描くプログラム

```
for( int i=0; i<10; i++){  
    描く図形（線、丸など）;  
}
```

```
for( int i=0; i<10; i++){  
    line(10,10,100,100);  
}
```

これだと同じ場所に描いてしまう。

○を100個描く3つの方法(その1)

○の場所を数値で座標指定

(これだと重ね描きになり1つの○にしか見えない)

```
for( int i=0; i<100; i++){
```

```
    ellipse(100,150,10,10);
```

```
}
```

半径

中心のY座標

中心のX座標

○を100個描く3つの方法(その2)

○の場所を整数変数stとenで座標指定

```
int st=10, en=20; //変数stとenに初期値を代入
```

```
for( int i=0; i<100; i++){
```

```
    ellipse( st, en, 10, 10);
```

```
    st= st + 5;
```

```
}
```

中心のX座標

中心のY座標

半径

○を100個描く3つの方法(その3)

○の場所を i を使って座標指定

(i は、繰り返しの度に1ずつ増える。初期値0)

```
for( int i=0; i<100; i++){
```

```
    ellipse( i , i*2 , 10 , 10 );
```

```
}
```

半径

中心のY座標


中心のX座標

○を100個描く3つの方法(その4)

○の場所を 乱数 を使って座標指定
(random(10)は、0~10 の乱数を作る)

```
for( int i=0; i<100; i++){  
    ellipse( random(5) , random(7), 10 , 10 );  
}
```

半径




中心のX座標

中心のY座標

色を変える

```
fill( 赤成分, 緑成分, 青成分 );
```

0 ~ 255



こんな風を書く

```
fill( 100, 200, 50 );
```

```
fill( i*5, i*20, random(100) );
```

```
fill( random(255), random(255), 0 );
```

○の縁線を消すにはnoStroke(); と書く

その他

- ○の縁線を消したければ、
`noStroke();` と書く。

今までの命令を組合わせた例

```
size(200,200);  
for( int i=0; i<100; i++){  
    noStroke();  
    fill(random(255), 100, i*2);  
    ellipse( random(200) , i*3, random(30) , 10 );  
}
```

さらに

- プログラム4: 先のプログラムをfor文で書くと、

```
int y= 0;
```

```
for(int i=0; i<5; i++){ //i++は、iに1を加えること
```

```
    line(10, y, 80, y);
```

```
    y= y+ 10; // yに10を加える (yは0,10,20,30,40と変わる)
```

```
}
```

- プログラム5: 変数iを使って同じことをやる

```
for(int i=0; i<5; i++){
```

```
    line( 10, i*10, 80, i*10); // *は掛け算を意味する、割り算は/
```

```
}
```

円だけ使ったデジタルアート

座標0, 0

```
size(200,200);  
で画面サイズ変更  
(初期値は100,100)
```