

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

準備 : step0: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  をクリックすると、



- step1:ウインドウ (左図の右部) 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。

### I. 変数と繰り返し for

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) ;

プログラム 2 : (変数<数値を入れる箱>を使う) プログラム 1 と同じことをやる :

```
int st=10, en=80; //プログラムで使う変数の宣言。 int は整数という意味。  
line(st, st, en, en); //プログラム本体
```

ここで 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 (<=) を単純化したもので、書込み<代入>を意味する)

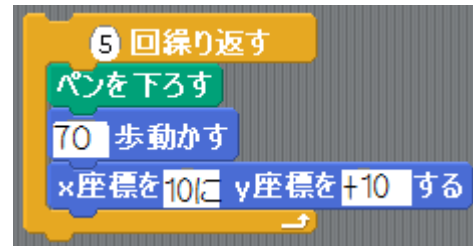
プログラム 3 : 5 本水平線を引く

```
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);
```



線の数が増えるとやってられないので、  
繰り返し命令を使う。

Scratch では「くり返す」だったが、  
プログラミング言語では  
「 for(...) {...} 」



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)
}
```

for(int i=0; i<5; i++){...} は{}の中のプログラムを 5 回繰り返すことを意味する。  
プログラム 4 では使っていないが、変数 i の数値は繰り返しのたびに 0 ~ 4 まで 1 ずつ増える。

プログラム 5 : 変数 i を使ってプログラム 3 と同じことをやる

```
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/
}
```

## I I . 線の色を変える

色は、R 赤 G 緑 B 青の 3 原色を数値 (各 0~255) で表現)

背景色を白にするには、

```
background(255,255,255);
```

線を黄色にするには、

```
stroke(255,255,0);
```

黄色のグラデーションを作る

```
background(255,255,255);
```

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

```
    stroke( 255, 255, i*2);
```

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

```
}
```



	<code>fill(255,0,0);</code>	赤
	<code>fill(0,255,0);</code>	緑
	<code>fill(0,0,255);</code>	青
	<code>fill(255,255,0);</code>	黄
	<code>fill(0,255,255);</code>	空色
	<code>fill(255,0,255);</code>	ピンク

ウインドウのサイズ変更は、

```
size(300,200); //size(横、縦);
```

I I I . 四角や丸を描く

四角を描くには、

```
rect(x, y, w, h); // (x, y)を左上角とし、指定した幅(w)と高さ(h)の長方形を描く
```

楕円を描くには、

```
ellipse(x, y, w, h); //x, y は円の中心座標。w,h は幅と高さ。
```

四角や円に色を塗るには、`fill( R, G, B );`

輪郭線を消すには `noStroke()`;

I V : 条件分岐

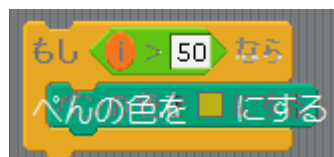
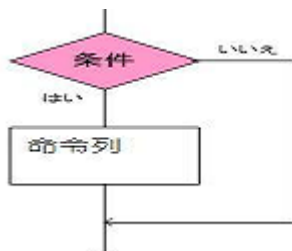
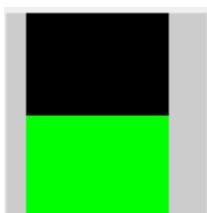
「もし～なら」ってやつ。

`if(...){...}`の(...)に示される条件が満たされた時だけ {...} を実行する。

例えば `if(i > 50){...}` は、もし `i` が 50 より大きければ {...}内を実行。

```
for(int i=0; i<100; i++){
  if( i>50){
    stroke(0, 255, 0);
  }
  line(10, i, 80, i);
}
```

色は黙っていると黒になるので、  
Y座標が 50 を超えたら、色を緑にするプログラム。

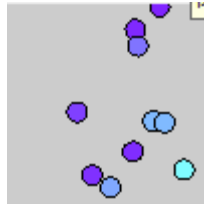


V : デジタルアートを作る

1. ランダムを色々な所に使うと、予期せぬ美が生まれる。

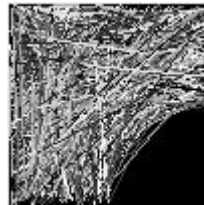
(1)ランダムな場所に、ランダムな色の丸を描く

```
for(int i=0; i<10; i++){  
    fill( 128, random(256), 255);  
    ellipse(random(100),random(100),  
        10, 10);  
}
```



(2)ランダムに線を引く

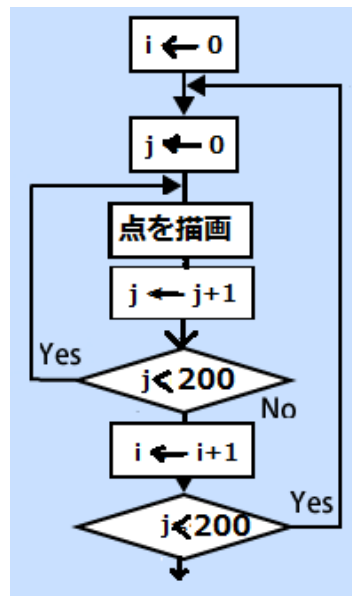
```
background(0,0,0);  
for(int i=0; i<2000; i++){  
    stroke(random(255));  
    line(random(50), random(100),  
        random(100), random(50));  
}
```



2. 二重の繰り返し for.for

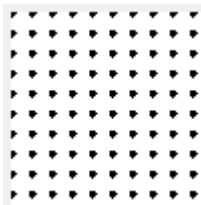
(1) 2次元平面のグラデーション

```
size( 200, 200);  
background( 255, 255, 255);  
noStroke();  
for(int i=0; i<200; i++){  
    for(int j=0; j<200; j++){  
        stroke( i, j, 0); //色の指定  
        point( i, j); //点を描く  
    }  
}
```



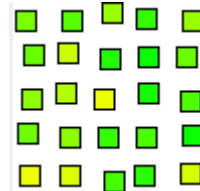
(2)小円の2次元配置

```
int x, y; //初期値なし
background( 255 ); //背景色の決定
fill(0); //黒で描画
for(int i = 0; i < 10; i++){ //縦の列の点を描画
  for(int j = 0; j < 10; j++){ //横の行の点を描画
    x= i * 10;
    y= j * 10;
    ellipse( x, y, 3, 3);
  }
}
```



(3) □のランダム配置

```
background(255,255,255);
for(int i=0; i<5; i++){
  for(int j=0; j<5; j++){
    fill(random(255),255,0);
    rect(i*20 + random(5),
      j*20 + random(5), 10, 10);
  }
}
```



## 付録:

その他の図形 :

三角形、triangle( x1, y1, x2, y2, x3, y3)

任意の四角形、quad( x1, y1, x2, y2, x3, y3, x4, y4)

円弧、arc( x, y, width, height, start, stop)

図形の塗りつぶし :

(白黒灰色の時)

fill(d) : d が 0 なら黒、255 なら白、その間の数は数が小さいほど濃い灰色になる

fill(d,a) : a は「透明度」で 0 から 255 の整数。0 は完全に透明、255 は完全に不透明。

(カラーの時)

fill(r,g,b) :

r,g,b はいずれも 0 から 255 までの整数で、255 がもっとも強い。r,g,b の値が同じなら灰色になる。

fill(r,g,b,a) : a は透明度。

図形の輪郭線 :

stroke(...) : 輪郭線の色を指定。指定のしかたは fill と同じ

strokeWeight(width) : 線の太さを指定。太さ(width)はピクセル単位

smooth(); //図形の輪郭をきれいに描画

strokeWeight(幅); // 線の太さ指定

画像の保存 : save ("screenshot.png");

画像を見るには「Sketch」メニューで「Show Sketch Folder」を選びフォルダーを開く。

参考サイト :

<http://www.cp.cmc.osaka-u.ac.jp/~kikuchi/kougi/simulation2009/processing0.html>

<http://www.geijutsu.tsukuba.ac.jp/~fumiaki/class2009/processing/term2/processing17.pdf>

<http://fry.no.coocan.jp/lecture/IS/procInt.pdf>