

Data Visualization WorkShop

October 17, 2016

Contents

Processing について

Processing の基本

Processing とは

Processing とはマサチューセッツ工科大学のメディアラボが開発したデザインや映像に特化したプログラミング開発環境.

Processing の特徴

- ▶ 命令がとても簡単
(<http://ossyaritoori.hatenablog.com/entry/2016/02/16/161753>)
- ▶ クロスプラットフォーム (Windows, MacOS, Linux, iOS などで作る)
- ▶ 様々な作品が生み出されている
(<http://www.creativeapplications.net/category/processing/>)

インストール

Processing のウェブサイトからダウンロードする
(<https://processing.org/download/>)

Processing にはアプリケーションに最初からサンプルコードがプリインストールされている「ファイル」>「サンプル...」

参考書籍

すでにたくさんの書籍が出版されています

- ▶ 「Processing をはじめよう」 オライリージャパン 2011 年
- ▶ 「Processing:ビジュアルデザイナーとアーティストのためのプログラミング入門」 BNN 新社 2015 年
- ▶ 「ジェネラティブ・アートーProcessing による実践ガイド」 BNN 新社 2014 年
- ▶ 「Generative Design ーProcessing で切り拓く、デザインの新たな地平」 BNN 新社 2016 年
- ▶ 「Nature of Code- Processing ではじめる自然現象のシミュレーション」 株式会社ボーンデジタル 2014 年
- ▶ 「ビジュアライジング・データーProcessing による情報視覚化手法」 オライリー・ジャパン 2008 年
- ▶ 「データ可視化プログラミング入門」 秀和システム 2013 年

Processing の基本

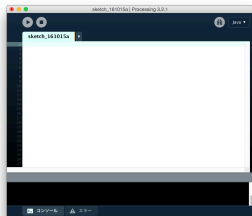


Figure 1: 画面



Figure 2: 結果

手順

- ▶ エディタでソースコードを書く
- ▶ Run ボタンでスケッチを実行する
- ▶ ウィンドウに描画内容が表示される
- ▶ コンソールにテキストメッセージやエラーコードが表示される
- ▶ Stop ボタンでスケッチを停止する

座標について

Processing では全て「ピクセル」で考えます.

円を描く

Examples/lecture/ellipse/ellipse.pde

```
ellipse(50, 50, 50, 20);
```

楕円（ellipse）を描く．

- ▶ 左から 50px
- ▶ 上から 50px
- ▶ 幅が 50px
- ▶ 高さが 20px

実行結果

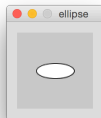


Figure 3: 楕円ができた

線を描く

Examples/lecture/line/line.pde

```
line(20, 20, 80, 80);
```

線 (line) を描く.

- ▶ 点 (20, 20) から
- ▶ 点 (80, 80) へ

実行結果

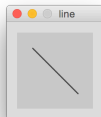


Figure 4: 線がひけた

三角形を描く

Examples/lecture/triangle/triangle.pde

```
triangle(10, 10, 90, 20, 60, 90);
```

三角形 (triangle) を描く.

- ▶ 点 (01, 10) と
- ▶ 点 (90, 20) と
- ▶ 点 (60, 90) の

実行結果

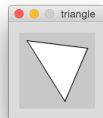


Figure 5: 三角形がかけた

四角形を描く

Examples/lecture/rect/rect.pde

```
rect(10, 10, 80, 80);
```

四角形（rect）を描く．

- ▶ 点 (10, 10) から
- ▶ 横 80px で
- ▶ 縦 80px の

実行結果

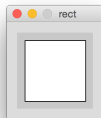


Figure 6: 四角形がかけた

画面サイズを変える

Examples/lecture/size/size.pde

```
size(500, 250);
```

画面サイズを変更する

- ▶ 横 500px
- ▶ 縦 250px に

実行結果

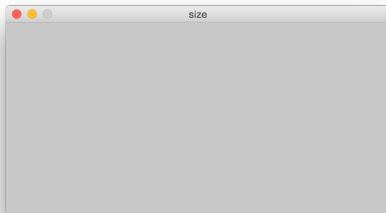


Figure 7: 出力サイズが変わった

線の太さを変える

Examples/lecture/strokeWeight/strokeWeight.pde

```
strokeWeight(5);  
ellipse(50, 50, 30, 30);
```

線の太さ（strokeWeight）を
変える.

- ▶ 線幅 5px で

実行結果

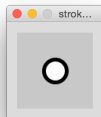


Figure 8: 線が太くなった

色を変える

Examples/lecture/color/color.pde

```
colorMode(HSB, 256);  
stroke(50, 255, 255);  
fill(122, 255, 255);  
background(200, 255, 255);  
strokeWeight(10);  
ellipse(50, 50, 80, 80);
```

色 (stroke, fill) を変える.

- ▶ 色空間を HSB で
- ▶ 線色を色彩 50, 彩度 255, 明度 255 で
- ▶ 塗り色を色彩 122, 彩度 255, 明度 255 で
- ▶ 背景を色彩 200, 彩度 255, 明度 255 で
- ▶ 線幅 10px で
- ▶ 点 (50, 50) から横 80px. 縦 80px の円

実行結果

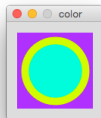


Figure 9: 色が変わった

HSB 色空間



Figure 10: Adobe Illustrator における HSB 設定

型

Examples/lecture/type01/type01.pde

```
int i;  
float f;  
String s;  
boolean b;  
  
i = 1;  
f = 0.1;  
s = "50";  
b = true;  
  
ellipse(s, s, s, s);
```

いろんな型

- ▶ 整数
- ▶ 少数
- ▶ 文字列
- ▶ 真偽
- ▶ 点 (50, 50) から横 50px, 縦 50px の円？

正しくは...

Examples/lecture/type02/type02.pde

```
int i;  
float f;  
String s;  
boolean b;  
  
i = 1;  
f = 0.1;  
s = "50";  
b = true;  
float fl = float(s);  
  
ellipse(fl, fl, fl, fl);
```

キャストとは

- ▶ 違う型にうつす
- ▶ 文字列 `s` を少数に
- ▶ 点 (50, 50) から横 50px, 縦 50px の円

実行結果



Figure 11: 円がかけた

条件文

Examples/lecture/joken/joken.pde

```
int a = 5;
int b = 4;
int c = 3;

if (a > b) println("a は b より大きいです");
if (a > b && a > c) println("a は b, c よりも大きいです");
if (a == b) println("a と b は同じです");
if (b >= c) println("b は c 以上です");
if (b >= c || a == c) println("b は c 以上, または a と c は同じです");
```

if 文

- ▶ 比較演算子いろいろで数字を比較
- ▶ if(xxxx) 文. xxx は真偽

繰り返し

Examples/lecture/loop/loop.pde

```
size(500, 100);  
for (int i = 0; i < 9; i++) {  
  ellipse(50 * (i + 1), 50, 50, 50);  
}
```

for 文

- ▶ 0 から 9 まで
- ▶ カッコ内の処理を繰り返す

実行結果

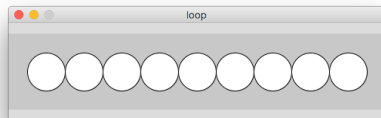


Figure 12: 円が 10 個かけた

配列

Examples/lecture/array/array.pde

```
int[] n = new int[5];

for (int i = 0; i < 5; i++) {
  n[i] = i * 100;
}

println(n[0]);
println(n[1]);
println(n[2]);
println(n[3]);
println(n[4]);
```

配列

- ▶ n という 5 つの箱（配列）を準備
- ▶ for 文のなかで，0 から 4 番目の箱に値を入れる

それではサンプルプログラムを見てみましょう.