

メディア技術基礎(web) 2012

- Processing の Library の読み込み方
- Twitter4j の使い方
- データ処理の基礎
- 情報のビジュアライゼーションについて

参考サイト

<http://yoppa.org/blog/3972.html>

Pachube

<http://www.pachube.jp/>

[Step1] Twitter4j の準備をします。

<https://docs.google.com/file/d/0BzyVHU69Q03mbVdxUGZsVE4yUFk/edit>

から、twitter4j.zip をダウンロード。

ふつうは、この Zip ファイルを解凍したライブラリを、
~/Document/Processing/libraries/ 以下にコピー。これで準備完了。
これが、Library の一般的なインストール手順です。

ただし、Processing を立ち上げた状態でこれを行っても反映されません。
必ず、Processing をいったん「すべて」とじ、そのあとで、必要ファイルをコピー
した後で、もういちど立ち上げなおしてください。

ただ、twitter4j は、ちょっとトリッキーなので別の方法で行います。
Zip ファイルを解凍したなかにある、twitter4j.jar を直接、Processing のソース
コードの上にドラッグ&ドロップするのです。
“One file added to Sketch” と表示されれば OK です。

[2] Twitter のメッセージを画面に表示させるスクリプト

```
import twitter4j.conf.*;
import twitter4j.internal.async.*;
import twitter4j.internal.org.json.*;
import twitter4j.internal.logging.*;
import twitter4j.http.*;
import twitter4j.internal.util.*;
import twitter4j.api.*;
import twitter4j.util.*;
import twitter4j.internal.http.*;
import twitter4j.*;

//Twitter
public Twitter myTwitter;
String twittername = "hiroyeah"; ///表示するツイート名
int countnum = 1000; ///表示するツイート数

String[] user = new String[countnum]; ///ユーザー名を格納
String[] msg = new String[countnum]; ///ツイートを格納
Date[] d = new Date[countnum]; /// 日付を格納
PFont myFont;

public void setup() {

    size(800, 600);
    background(0, 0, 0);
    readtwitter();
    myFont = loadFont("MS-PGothic-10.vlw");
};

void draw() {
    background(0);
    textFont(myFont, 10);
    for (int i=0; i < countnum; i++){
```

```

        text("Tweet by " + user[i] + " at " + msg[i] + " in " + d[i], 10, i*10);
    }

};

public void readtwitter() {
    myTwitter = new Twitter("", "");
    try {
        Query query = new Query();
        query.setQuery("from:"+twittername);
        query.setRpp(countnum);
        QueryResult result = myTwitter.search(query);
        ArrayList tweets = (ArrayList) result.getTweets();

        for (int i = 0; i < tweets.size(); i++) {
            Tweet t = (Tweet) tweets.get(i);
            user[i] = t.getFromUser();
            msg[i] = t.getText();
            d[i] = t.getCreatedAt();

            println("Tweet by " + user[i] + " at " + msg[i] + " in " + d[i]);
        };

    }
    catch (TwitterException te) {
        println("Couldn't connect: " + te);
    };
}

```

説明のポイント

- (1) 関数自体をつくること readtwitter();
- (2) 配列の復習
- (3) テキスト描画の復習 printlnによる確認

[2b] Twitter の借用

センサーデータ流通とセンサーデータビジュアライゼーション

<http://www.pachube.jp/>



Pachube使用画面



Pachube.com - Home | Pachube.com - Home | Pachube.com - Home

Pachubeの使い方

Pachube.com - Home | Pachube.com - Home | Pachube.com - Home

- Java
- Processing
- Ruby
- PHP



[3] Twitter に流れてくるセンサデータを画面に表示させるスクリプト

```
import twitter4j.conf.*;
import twitter4j.internal.async.*;
import twitter4j.internal.org.json.*;
import twitter4j.internal.logging.*;
import twitter4j.http.*;
import twitter4j.internal.util.*;
import twitter4j.api.*;
import twitter4j.util.*;
import twitter4j.internal.http.*;
import twitter4j.*;

//Twitter
public Twitter myTwitter;
// いま稼働しているのは sensorbot0005

String twittername = "sensorbot0005";
int countnum = 1000;

float[] pr0 = new float[countnum];
float[] pr1 = new float[countnum];
float[] pr2 = new float[countnum];
float[] pr3 = new float[countnum];

int time = 100;
float scale0 = 0.5;
float scale1 = 0.5;
float scale2 = 0.5;
float scale3 = 0.5;
```

```

public void setup() {

    size(800, 600);
    background(0, 0, 0);
    readtwitter();

};

void draw() {

    background(0);

    /// param1
    /// Auto-Scaling
    float pmax = 0; float pmin = 0;
    for (int i=0; i<time-1; i++) {
        pmax = max(pr0[i], pmax);
        pmin = min(pr0[i], pmin);
    }

    int starty = 100;
    int endy = 600;
    int startx = 100;
    int endx = 700;

    int stepx = (endx- startx)/ time;
    float stepy = ((endy - starty)/(pmax - pmin));
    println(""+stepy);

    for (int i=0; i<time-1; i++) {
        stroke(255, 0, 0);
        line(startx + i*stepx, endy - pr0[i]*scale0*stepy, startx + (i+1)*stepx,
endy - pr0[i+1]*scale0*stepy);
    }
}

```

```

    /// param2
    /// Auto-Scaling
    pmax = 0; pmin = 0;
    for (int i=0; i<time-1; i++) {
        pmax = max(pr1[i], pmax);
        pmin = min(pr1[i], pmin);
    }

    starty = 100;
    endy = 600;

    stepx = (endx- startx)/ time;
    stepy = ((endy - starty)/(pmax - pmin));

    for (int i=0; i<time-1; i++) {
        stroke(0, 255, 0);
        line(startx + i*stepx, endy - pr1[i]*scale1*stepy, startx + (i+1)*stepx,
endy - pr1[i+1]*scale1*stepy);
    }

    /// param3
    /// Auto-Scaling
    pmax = 0; pmin = 0;
    for (int i=0; i<time-1; i++) {
        pmax = max(pr2[i], pmax);
        pmin = min(pr2[i], pmin);
    }

    starty = 100;
    endy = 600;

    stepx = (endx- startx)/ time;
    stepy = ((endy - starty)/(pmax - pmin));

```



```

for (int i=0; i<time-1; i++){
    stroke(0, 0, 255);
    line(startx + i*stepx, endy - pr2[i]*scale2*stepy, startx + (i+1)*stepx,
endy - pr2[i+1]*scale2*stepy);
}

/// param4

/// Auto-Scaling
pmax = 0; pmin = 0;
for (int i=0; i<time-1; i++){
    pmax = max(pr3[i], pmax);
    pmin = min(pr3[i], pmin);
}

starty = 100;
endy = 600;

stepx = (endx- startx)/ time;
stepy = ((endy - starty)/(pmax - pmin));

for (int i=0; i<time-1; i++){
    stroke(255, 255, 0);
    line(startx + i*stepx, endy - pr3[i]*scale3*stepy, startx + (i+1)*stepx,
endy - pr3[i+1]*scale3*stepy);
}

};

public void readtwitter() {
    myTwitter = new Twitter("", "");
    try {

```

```

Query query = new Query();
query.setQuery("from:"+twittername);
query.setRpp(countnum);
QueryResult result = myTwitter.search(query);

ArrayList tweets = (ArrayList) result.getTweets();

for (int i = 0; i < tweets.size(); i++) {
    Tweet t = (Tweet) tweets.get(i);
    String user = t.getFromUser();
    String msg = t.getText();
    Date d = t.getCreatedAt();
    println(msg);
    String[] p = splitTokens(msg, " ");
    String[] pp0 = splitTokens(p[0], " ");
    String[] pp1 = splitTokens(p[1], " ");
    String[] pp2 = splitTokens(p[2], " ");
    String[] pp3 = splitTokens(p[3], " ");

    pr0[i] = float(pp0[1]);
    pr1[i] = float(pp1[1]);
    pr2[i] = float(pp2[1]);
    pr3[i] = float(pp3[1]);

    println("Tweet by " + user + " at " + d + ": " + msg);
    println(pr0[i] + " ... " + pr1[i] + " ... " + pr2[i] + " ... " + pr3[i]);
};

}
catch (TwitterException te) {
    println("Couldn't connect: " + te);
};
}

```

説明のポイント

- (1) splitTokens(msg, ",");
- (2) 「型」の整理。Float() テキスト→数値

2週間後までの課題 Processing 最終課題

Twitter 名「sensorbot0005」に流れてくる、農業用センサからのデータを、Processing の 2 次元（場合によっては 3 次元）描画機能を用いて、直感的にわかりやすいグラフィックスとして表示してください。マウスとのインタラクション、より高度な処理など、ここまで学んだ Processing のすべての知識をつかってよいです。（もちろん学んでいないことも、自分で調べてつかってみてよいです）。

情報可視化 (Information Visualization)

<http://gihyo.jp/dev/feature/01/visualization/0001>

ビジュアライジング・データ —Processing による情報視覚化手法

<http://www.amazon.co.jp/%E3%83%93%E3%82%B8%E3%83%A5%E3%82%A2%E3%83%A9%E3%82%A4%E3%82%B8%E3%83%B3%E3%82%B0%E3%83%BB%E3%83%87%E3%83%BC%E3%82%BF-%E2%80%95Processing%E3%81%AB%E3%82%88%E3%82%8B%E6%83%85%E5%A0%B1%E8%A6%96%E8%A6%9A%E5%8C%96%E6%89%8B%E6%B3%95-Ben-Fry/dp/4873113784>

ツラグラ

<http://www.tsutagra.go.jp/>

http://blog.livedoor.jp/ld_directors/archives/51686376.html

きれいで、わかりやすく、みやすい、インフォグラフィクスをつくりましょう！

（次週は、特別対論です。Mozilla Japan の赤塚大典さん(SFC 卒)をお呼びして、マークアップエンジニア、ウェブデザインの現在の仕事や状況についてお聞きします）。