

2014 年度情報メディア基盤ユニット参考資料その 1 落ち葉拾い：ファイルからのデータの読み出し

Processing 言語では、ファイルからのデータの読み出しや、書き込みを行うことができます。Processing 言語のバージョンが 2.2 以上の場合には、ファイルからのデータの読み出しや書き出しを行う関数が含まれています。しかし、バージョンが 1.5 の場合には、それほど多くはありません。ここでは、どちらでも使える、基本的な関数を紹介します。

ファイルからのデータの読み出しには、

- 1) 一気に毎行に全てのデータを読み出す、
- 2) 一行ずつデータを読み出す、

の 2 通りの方法があります。

一番簡単の方法は、1)の方法です。つぎのサンプルは”test.txt”というファイルからデータを読み出し、それを表示するプログラムです。

サンプル

```
1 String[] moji;  
2 PFont font;  
3  
4 void setup(){  
5     size(400,200);  
6     font = loadFont("Serif-48.vlw");  
7     textFont(font,36);  
8     moji = loadStrings("text.txt"); // text.txt ファイルからデータを読み出す。  
9 }  
10  
11 void draw(){  
12     background(255);  
13     fill(0);  
14     textAlign(CENTER,CENTER);  
15     text(moji[second() % moji.length],0,0,width,height);  
16 }
```

text.txt の内容

```
1 Kongo  
2 Hiei  
3 Haruna  
4 Kirishima  
5 Fuso  
6 Yamashiro  
7 Ise  
8 Hyuga  
9 Nagato  
10 Mutsu  
11 Yamato  
12 Musashi
```

ファイルからデータを読み出すために、loadStrings 関数を使用しています。この関数は、ファイルからデータを読み出し、そのデータを String 型の配列に保存するような関数です。このサンプルでは、text.txt というファイルからデータを読み出し、このファイルのデータを moji という String 型の配列に保存します。moji[0]には、ファイルの 1 行目のデータが文字列として記録されています。同様に、moji[1]にはファイルの 2 行目、moji[2]にはファイルの 3 行目の情報が記録されています。ファイルの最後の行のデータは moji[moji.length-1]に記録されています。loadStrings 関数では、ファイルのデータを一気に読み出し、メモリ中に保存するので、余り大きなファイルを読み込むことは難しいと思います。

String[] loadStrings (ファイル名)

ファイル名で指定されたファイルからデータを読み出し、その結果を String 型の配列変数に保存する。読み出された情報は String 型となっています。また、一般的には、"Show sketch folder"で表示されるフォルダ内にある data フォルダに保存されているファイルからデータを読み出します。つまり、読み出したファイルはこの data フォルダに保存しておくことが必要となります。

loadStrings 関数を使った方法では、ファイルから読み出したデータはすべて文字列(String) となってしまいます。数値データが入っている場合には、どのようにしたら良いでしょうか？ 一番単純な方法は、文字列として読み込んだデータを int 型や float 型に変換する方法です。次のサンプルでは、loadStrings 関数を使って fuel.txt というファイルからデータを読み出し、そのデータを int 関数によって、int 型の値に変換しています。

サンプル

```
1 String[] moji;
2 int[] fuel;
3 int barWidth = 20;
4 void setup(){
5     moji = loadStrings("fuel.txt");
6     fuel = new int[moji.length];
7     for(int i=0;i < moji.length;i++){
8         fuel[i] = int(moji[i]);
9     }
10    size(moji.length*barWidth,max(fuel));
11 }
12
13 void draw(){
14     background(255);
15     stroke(255);
16     for(int i=0;i<fuel.length;i++){
17         fill(255-fuel[i]);
18         rect(barWidth*i,height-fuel[i],barWidth,fuel[i]);
19     }
20 }
```

fuel.txt の内容

1	80
2	80
3	80
4	80
5	85
6	85
7	85
8	85
9	100
10	100
11	250
12	250

このサンプルでは、メモ帳などで読み込みに使うデータを作っていました。ゲームなどを作る際には、Excel などを利用してデータファイルを作成することがあります。拡張子が“.xlsx”などのファイルを読み込むことは面倒ですが、拡張子が“.csv”というファイルでは、割と簡単にデータを読み出すことができます。

次のような表の情報は

name	fuel	durableness
Kongo	80	63
Hiei	80	63
Haruna	80	63
Kirishima	80	63
Fuso	85	67
Yamashiro	85	67
Ise	85	74
Hyuga	85	74
Nagato	100	80
Mutsu	100	80
Yamato	250	93
Musashi	250	94

“.csv”ファイル(“battleships.csv”)では、

```
1 name,fuel,durableness
2 Kongo,80,63
3 Hiei,80,63
4 Haruna,80,63
5 Kirishima,80,63
6 Fuso,85,67
7 Yamashiro,85,67
8 Ise,85,74
9 Hyuga,85,74
10 Nagato,100,80
11 Mutsu,100,80
12 Yamato,250,93
13 Musashi,250,94
```

のように、1行の情報が”,”で区切られて並んでいます。例えば、このファイル(battleships.csv)の情報を loadStrings 関数で読み込む (lines = loadStrings(“battleships.csv”);) とすると、lines[0]は”name,fuel,durableness”という文字列が記録されることになります。

同様に、lines[1]は”Kongo,80,63”、lines[2]は”Hiei,80,63”、lines[3]は”Haruna,80,63”、lines[12]は”Musashi,250,94”という文字列が記録されることになります。配列 lines に記録されている情報は、名前,燃料,耐久という情報が並んでいます。つまり、”,”で区切られている一つ一つの項目に意味があります。そこで、この意味のある情報をバラバラにして取り出すことが必要になります。このような目的に使える関数として split 関数があります。この関数は文字列を指定した文字を区切り文字を基準に分割する関数です。

String[] split(文字列,区切り文字)

文字列を区切り文字で分割し、その結果を String 型の配列に保存する関数です。

例えば、String[] items = split(“Kongo,80,63”,”,”)とすると、“Kongo,80,63”という文字列を区切り文字”,”で分割するので、items[0]には”Kongo”、items[1]には”80”、items[2]には”63”という文字列が保存されることになります。

この利用したサンプルを次に示します。

サンプル

```
1 PFont font;
2
3 String[] name;
4 int[] fuel;
5 int[] durable;
6
7 int yourID;
8 int myID;
9 boolean playing = false;
10
```

```
11 void setup(){
12     size(300,210);
13     String[] lines = loadStrings("battleships.csv");
14     font = loadFont("Serif-48.vlw");
15     textFont(font,24);
16
17     name = new String[lines.length-1];
18     fuel = new int[lines.length-1];
19     durable = new int[lines.length-1];
20     for(int i=0;i<lines.length-1;i++){
21         String[] items = split(lines[i+1],",");
22         println(items);
23         name[i] = items[0];
24         fuel[i] = int(items[1]);
25         durable[i] = int(items[2]);
26     }
27
28 }
29
30 void draw(){
31     background(255);
32     fill(0);
33     if(playing){
34         textAlign(CENTER,CENTER);
35         text("You choose "+name[yourID]+".",0,0,width,70);
36         text("I choose "+name[myID]+".",0,70,width,70);
37         String msg;
38         if(durable[myID] > durable[yourID]){
39             msg = "I win!!";
40         }else if(durable[myID] == durable[yourID]){
41             msg = "Draw!!";
42         }else{
43             msg = "You win!!";
44         }
45         text(msg,0,140,width,70);
46     }
47 }
48
49 void mouseClicked(){
50     yourID = int(random(name.length));
51     myID = int(random(name.length));
52     playing = true;
53 }
```

このように loadStrings 関数などを使うとファイルからデータを読み出すことが出来るようになります。