



プログラミング演習2 ファイル入出力

中村, 小松, 小林, 鹿喰

データの読み出し



• ファイルから読み込むとは？

C:¥ どこかのフォルダ **¥** どこかのフォルダ **¥** なんかのファイル

上から順に読み込む

こんにちは みさなん おんげき ですか？

わたしは げんき です。

この ぶんしょう は いりぎす の ケブンツリジ
だがいくの けゆきんう の けっか

にんんげ は もじを にしんき するとき その
さしいよとさいご の もさじえ あいてつれば

じばんゆん は めくちちゃや でもちんやと

よめる という けゆきんう にもづいとて わざと
もじの じんばゆん を いかれえて あまりす。

どでうす？ ちんやとよやちめうでしょ？

ちんやとよためら はゆしくを よしろく

記録したり呼び出したり



- これまで

- 変数に値を保存(代入)し, 変数を使うことで値を取り出し表示したりしていた
 - Ballのx, y座標と, スピード
 - ゲームのscore
 - 電光掲示板のライトのON/OFF情報

アプリケーションを再起動したら消えてしまう



ファイルに記録して再起動後にも使う!

ファイルに保存する



- 現在の状態をファイルに保存
 - $x = 10$
 - $y = 20$
- ファイル (data.txt) に例えば下記のように保存

10

20

- 1行目は x , 2行目は y の値として読み込む

ファイルから読み込む



```
String [] lines = loadStrings("ファイル名");
```

- ファイルの中身を1行毎にStringの配列に格納
 - 1行目の値は lines[0] に, 2行目の値は lines[1] に入っている
- int(文字列) で, 文字列を整数に変換

```
String [] lines = loadStrings("data.txt");  
int x = int( lines[0] );  
int y = int( lines[1] );
```



```
saveStrings("ファイル名", String型の配列);
```

- String型の配列の内容を, 1行毎にファイルに書き込む
 - 例えば, Stringの配列をlinesとしたときに, lines[0]は1行目に, lines[1]は2行目に保存される

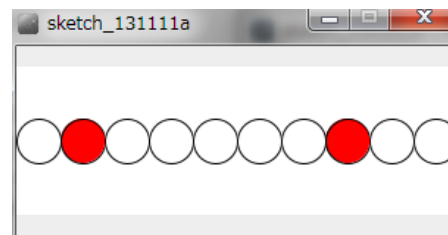
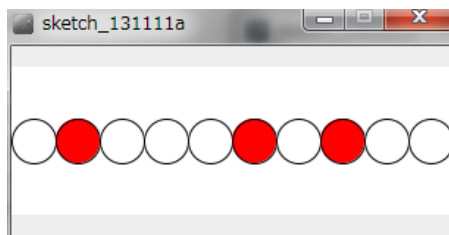
電光掲示板プログラム



(Q1) 横に10個，縦に1個並んだ電光掲示板(画面サイズは300x100とする)について，丸をクリックする度に赤色，白色と塗りつぶしの入れ替わるようにせよ.

• ヒント

- 整数型の配列lights(要素数は10個)を作成し，値が1なら赤色，0なら白色となるようにする
- マウスクリックされた位置にあるlightsの値を，1から0または0から1に変更する





```
int [] lights = new int [10];
void setup() {
  size( 300, 100 );
  int i=0;
  while( i<10 ){
    lights[i] = 0;
    i++;
  }
}

void draw() {
  background( 255 );
  int i=0;
  while( i<10 ){
    if ( lights[i] == 1 ) {
      fill( 255, 0, 0 );
    }
    else {
      fill( 255 );
    }
    ellipse( 15+30*i, 50, 30, 30 );
    i++;
  }
}
```

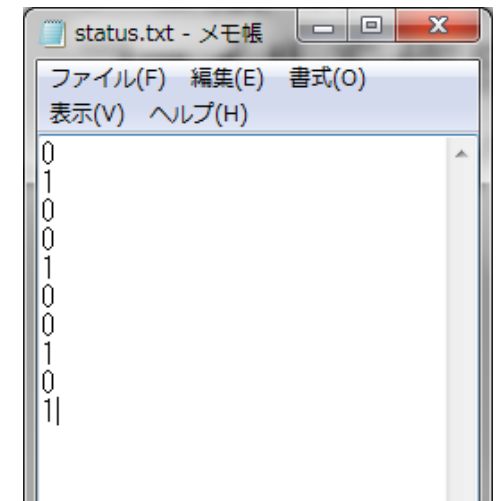
```
void mousePressed() {
  int i=0;
  while( i<10 ){
    if( dist(15+30*i, 50, mouseX, mouseY) <=15 ) {
      if( lights[i] == 1 ) {
        lights[i] = 0;
      } else {
        lights[i] = 1;
      }
    }
    i++;
  }
}
```


電光掲示板プログラム



(Q2) Q1を変更し, ファイルから赤白の状態を読み込むようにせよ

- メモ帳で, 1行に0または1だけを書いた10行のファイルを作成する(ファイル名はstatus.txt)
 - PDEと同じフォルダに保存する
- status.txt をプログラムで読み込み, 1行目をlights[0]に, 2行目をlights[1]にと値を順に割り当てる
- lightsの値に応じて描画

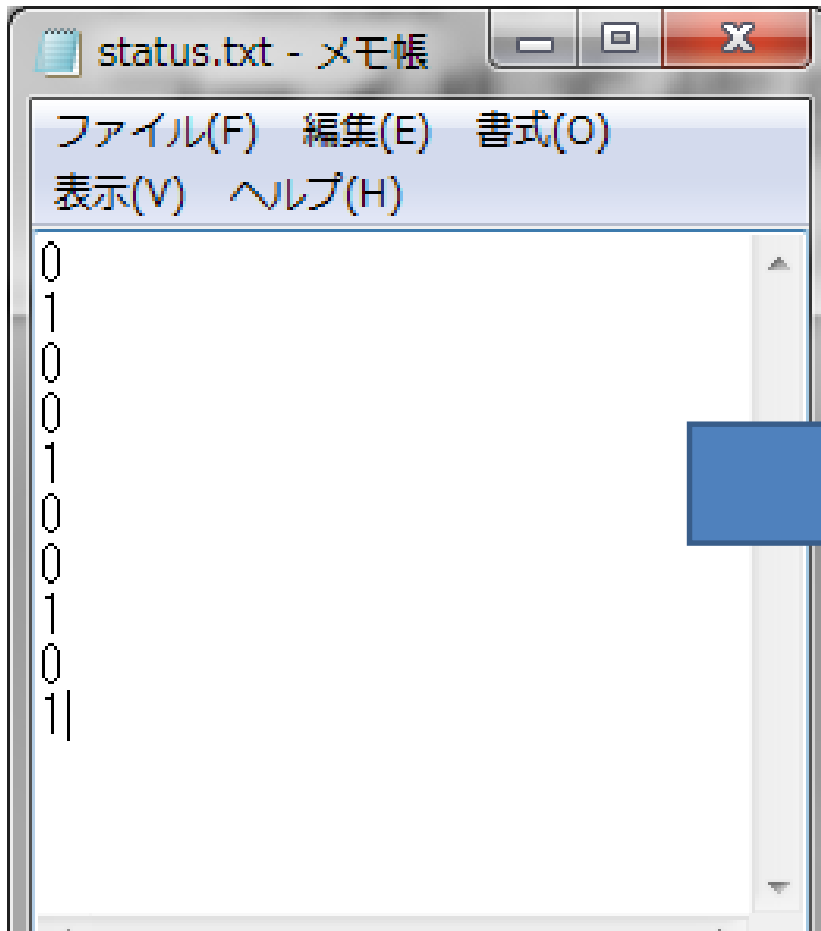


電光掲示板プログラム



- ファイルから読み込む

```
String [] lines = loadStrings("status.txt");
```



配列	値
lines[0]	"0"
lines[1]	"1"
lines[2]	"0"
lines[3]	"0"
lines[4]	"1"
lines[5]	"0"
lines[6]	"0"
lines[7]	"1"
lines[8]	"0"
lines[9]	"1"


電光掲示板プログラム

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室



setup()のみを変更してファイルから読み込む！

```
int [] lights = new int [10];  
void setup() {  
  size( 300, 100 );  
  String [] lines = loadStrings( "status.txt" );  
  lights[0] = int( lines[0] );  
  lights[1] = int( lines[1] );  
  lights[2] = int( lines[2] );  
  lights[3] = int( lines[3] );  
  lights[4] = int( lines[4] );  
  lights[5] = int( lines[5] );  
  lights[6] = int( lines[6] );  
  lights[7] = int( lines[7] );  
  lights[8] = int( lines[8] );  
  lights[9] = int( lines[9] );  
}
```



```
int [] lights = new int [10];  
void setup() {  
  size( 300, 100 );  
  String [] lines = loadStrings( "status  
  int i=0;  
  while( i<10 ) {  
    lights[i] = int( lines[i] );  
    i++;  
  }  
}
```

こんなエラーが出たら



- status.txt がPDEと同じフォルダに入っていないということ

The screenshot shows the Processing IDE window titled "lights_savveload | Processing 2.0.3". The code in the editor is as follows:

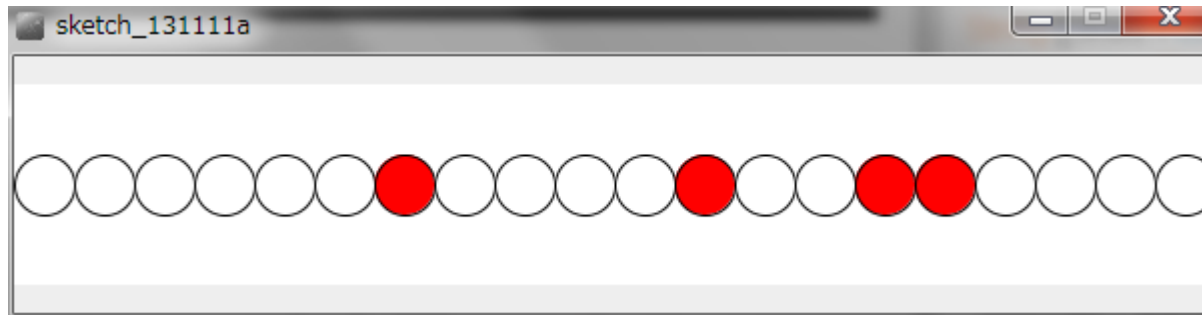
```
int [] lights = new int [20];
void setup() {
  size( 600, 100 );
  String [] lines = loadStrings( "status.txt" );
  lights[0] = int( lines[0] );
  lights[1] = int( lines[1] );
  lights[2] = int( lines[2] );
  lights[3] = int( lines[3] );
  lights[4] = int( lines[4] );
  lights[5] = int( lines[5] );
  lights[6] = int( lines[6] );
  lights[7] = int( lines[7] );
  lights[8] = int( lines[8] );
  lights[9] = int( lines[9] );
  lights[10] = int( lines[10] );
  lights[11] = int( lines[11] );
  lights[12] = int( lines[12] );
  lights[13] = int( lines[13] );
  lights[14] = int( lines[14] );
  lights[15] = int( lines[15] );
  lights[16] = int( lines[16] );
  lights[17] = int( lines[17] );
  lights[18] = int( lines[18] );
  lights[19] = int( lines[19] );
}
```

The line `lights[0] = int(lines[0]);` is highlighted in yellow. Below the code editor, a status bar shows "Done Saving." and a red error message: "The file "status.txt" is missing or inaccessible, make sure the URL is valid or that the file has been added to your sketch and is readable." The page number "5" is visible in the bottom left corner.

演習 (提出: lights_load)



- status.txtの値を変更するだけでプログラムの表示結果が変わることを確認せよ
- 電光掲示板を縦に1, 横に20に変更し, status.txtも変更して実行せよ





(Q3) Q2を変更し, クリックする度に赤白の状態をファイル(status.txt)に保存するようにせよ

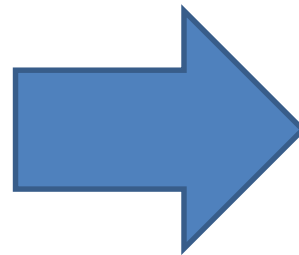
- 保存するためのString型の配列を作成する
 - `String [] lines = new String [10];`
 - 必要な行数分, String型の配列を用意する
 - `lines[i] = str(lights[i]);`
 - `str` で整数型をString型に変換する
- `saveStrings`で保存する

電光掲示板プログラム



```
saveStrings("status.txt", lines);
```

配列	値
lines[0]	"0"
lines[1]	"1"
lines[2]	"0"
lines[3]	"0"
lines[4]	"1"
lines[5]	"0"
lines[6]	"0"
lines[7]	"1"
lines[8]	"0"
lines[9]	"1"



```
status.txt - メモ帳  
ファイル(F) 編集(E) 書式(O)  
表示(V) ヘルプ(H)  
0  
1  
0  
0  
1  
0  
0  
1  
0  
1
```

電光掲示板プログラム

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室



```
void mousePressed() {
    int i=0;
    while( i<10 ) {
        if( dist(15+30*i, 50, mouseX, mouseY) <=15 ) {
            if( lights[i] == 1 ) {
                lights[i] = 0;
            } else {
                lights[i] = 1;
            }
        }
        i++;
    }

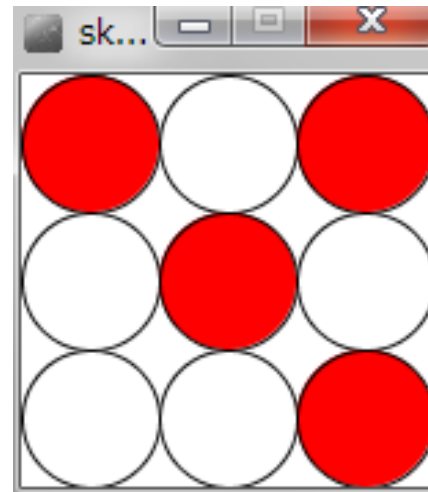
    i=0;
    String [] lines = new String [10];
    while( i<10 ) {
        lines[i] = str( lights[i] );
        i++;
    }
    saveStrings( "status.txt", lines );
}
```


演習 (lights_3x3)



(Q4) Q3の電光掲示板のプログラムを改良して縦に3個, 横に3個となるように変更し, 左下のよう
に記述したファイルから状態を読み込み下図のよう
に出力せよ. また, 値を変えて動作を確認せよ

```
1, 0, 1
0, 1, 0
0, 0, 1
```



演習 (lights_3x3)



ヒント

- 3x3の配列を作る
- 1行ずつ取得し, そのそれぞれの値を配列に代入していく
- 1行の値を読み込む時は, カンマ区切りであることを考慮
 - カンマごとに区切る場合は `split` を使う

駅名 + 値段を表示する



(Q5) 駅名 + 値段を表示するプログラムを作成する。なお、駅名と値段の情報は list.txt から読み込むものとする

Ochanomizu	130
Iida-bash	150
Shinjuku	190
Nakano	210
Kichijoji	280

ファイルを読み込む



list.txt (駅名, 値段が各行に)

Ochanomizu

130

lida-bash

150

Shinjuku

190

Nakano

210

Kichijoji

280

A screenshot of a terminal window titled "ticket_bending_machine". The window displays the contents of a file named "list.txt", which lists station names and their corresponding ticket prices. The text is as follows:

Ochanomizu	130
lida-bash	150
Shinjuku	190
Nakano	210
Kichijoji	280

```
String [] names = new String [5];  
int [] prices = new int [5];
```

```
void setup() {  
    size( 400, 400 );  
    String [] lines = loadStrings( "list.txt" );  
    textSize( 30 );  
    names[0] = lines[0];  
    prices[0] = int(lines[1]);  
    names[1] = lines[2];  
    prices[1] = int(lines[3]);  
    names[2] = lines[4];  
    prices[2] = int(lines[5]);  
    names[3] = lines[6];  
    prices[3] = int(lines[7]);  
    names[4] = lines[8];  
    prices[4] = int(lines[9]);  
}
```

```
void draw() {  
    background(255);  
    fill(0);  
    for ( int i=0; i<5; i++ ) {  
        text( names[i], 50, i*50+50 );  
        text( prices[i], 300, i*50+50 );  
    }  
}
```



ファイルを読み込む



```
String [] names = new String [5];  
int [] prices = new int [5];
```

```
void setup() {  
    size( 400, 400 );  
    String [] lines = loadStrings( "list.txt" );  
    textSize( 30 );  
    for( int i=0; i<5; i++ ){  
        names[i] = lines[i*2];  
        prices[i] = int(lines[i*2+1]);  
    }  
}
```

```
void draw() {  
    background(255);  
    fill(0);  
    for ( int i=0; i<5; i++ ) {  
        text( names[i], 50, i*50+50 );  
        text( prices[i], 300, i*50+50 );  
    }  
}
```

ファイルを読み込む



- ファイルを修正して1行を1つの駅名と値段に

Ochanomizu

130

lida-bash

150

Shinjuku

190

Nakano

210

Kichijoji

280

list.txt



Ochanomizu,130

lida-bash,150

Shinjuku,190

Nakano,210

Kichijoji,280

list.csv

カンマ区切りで書く

ファイルを読み込む



```
String [] names = new String [5];
int [] prices = new int [5];

void setup() {
  size( 400, 400 );
  String [] lines = loadStrings( "list.csv" );
  textSize( 30 );
  for( int i=0; i<5; i++ ){
    String [] data = split( lines[i], ',' );
    names[i] = data[0];
    prices[i] = int( data[1] );
  }
}
```

`String [] 文字列配列 = split(文字列, '区切り文字');`



```
String [] names;  
int [] prices;
```

```
void setup() {  
  size( 400, 400 );  
  String [] lines = loadStrings( "list.csv" );  
  names = new String [lines.length];  
  prices = new int [lines.length];  
  textSize( 30 );  
  for( int i=0; i<lines.length; i++ ) {  
    String [] data = split( lines[i], ',' );  
    names[i] = data[0];  
    prices[i] = int( data[1] );  
  }  
}
```

String [] 文字列の配列
= split(文字列, '区切り文字');

**配列.length で
配列のサイズ**

```
void draw() {  
  background(255);  
  fill(0);  
  for ( int i=0; i<names.length; i++ ) {  
    text( names[i], 50, i*50+50 );  
    text( prices[i], 300, i*50+50 );  
  }  
}
```



- list.csv ファイルを書き換えて、値を追加したり、色々な値を読み込んでみましょう

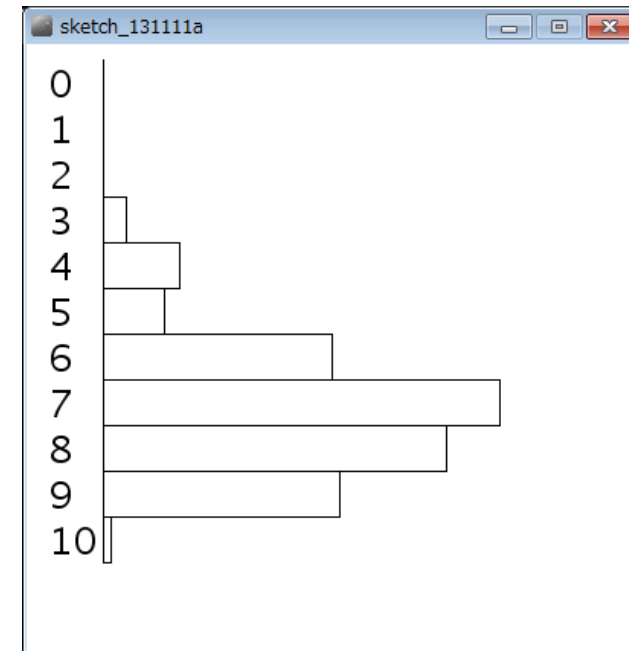
```
Ochanomizu,130  
lida-bash,150  
Shinjuku,190  
Nakano,210  
Kichijoji,280  
Ogikubo,290  
Mitaka,380  
Kokubunji,540  
Nishi-kokubunji,540  
Tachikawa,620
```

Ochanomizu	130
lida-bash	150
Shinjuku	190
Nakano	210
Kichijoji	280
Ogikubo	290
Mitaka	380
Kokubunii	540

おまけ: 成績グラフ作成



- <http://nkmr.io/lecture/2014/scores.txt> という180人分の成績データ(架空)を読み込んで、点数毎の頻度分布を作ってみましょう
 - 0-10までの数を数える配列を作成する
 - ファイルから読み込んだ値に応じて、配列のそれぞれの数を1ずつ増やしていく



ネット上から読み込む



```
String [] lines = loadStrings("URL");
```

```
int [] scores = new int [11];  
void setup(){  
  size( 400, 400 );  
  textSize( 25 );  
  for( int i=0; i<11; i++ ){  
    scores[i] = 0;  
  }  
  String [] lines  
    = loadStrings( "http://nkmr.io/lecture/2014/scores.txt");  
  for( int i=0; i<lines.length; i++ ){  
    scores[ int( lines[i] ) ]++;  
  }  
}
```

```
http://nkmr.io/lecture/2014/saikoro10.csv  
http://nkmr.io/lecture/2014/saikoro1000.csv  
http://nkmr.io/lecture/2014/saikoro100000.csv  
http://nkmr.io/lecture/2014/saikoro10000000.csv
```

おまけ: 簡易描画アプリ作成

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室



- クリックした場所に小さな円を描く。また、描いた情報をファイルに記録し、再現可能とする



```
class Dot {
    int x;
    int y;
    Dot( int _x, int _y ){
        x = _x;
        y = _y;
    }
    void paint() {
        ellipse( x, y, 3, 3 );
    }
}

ArrayList<Dot> dots;
void setup() {
    size( 400, 400 );
    dots = new ArrayList<Dot>();
    loadFromFile();
}

void draw() {
    background( 255 );
    for( int i=0; i<dots.size(); i++ ) {
        Dot dot = dots.get(i);
        dot.paint();
    }
}
```

```
void mousePressed() {
    Dot d = new Dot( mouseX, mouseY );
    dots.add(d);
    saveToFile();
}

void saveToFile() {
    String [] lines = new String [dots.size()];
    for( int i=0; i<dots.size(); i++ ) {
        Dot dot = dots.get(i);
        lines[i] = dot.x + "," + dot.y;
    }
    saveStrings("dot.csv", lines );
}

void loadFromFile() {
    String [] lines = loadStrings( "dot.csv" );
    for( int i=0; i<lines.length; i++ ) {
        String [] data = split( lines[i], ',' );
        Dot d = new Dot( int(data[0]), int(data[1]) );
        dots.add( d );
    }
}
```

演習(提出: lights10x20)



- 縦10x横20の電光掲示板を作成せよ. また, クリックによって色を変更するようにせよ.
 - 色は, 白→赤→白→赤と変化するようにせよ
 - 状態をファイルに保存し, 表示内容を保存および再生できるようにせよ
 - 自身の姓(漢字またはひらがな, カタカナなど)を描き, 起動した際に読み込めるようにせよ

演習 (提出: click_count)



ウィンドウ上でクリックした回数を表示するプログラムを作成せよ. なお, クリックした回数を保存し, 次回起動した時にそのクリック回数から増やしていくようにせよ

- count.txt というファイルを作成
- click_count という変数を用意して, mousePressed() の度に値を増やす
- mousePressed() の度に, click_count の値を count.txt に保存する
- 起動時 (setup) の時に, count.txt の値を click_count に読み込む