



プログラミング演習2

ファイル入出力

高橋, 中村, 小林, 橋本

今日のスケジュール



- 13:30-14:00 試験の解説 (ざっと)
- 14:00-15:00 今日の解説
- 15:00-16:45 課題に取り組む時間
- 16:45-17:00 課題解説の時間



- 最終回はグループワーク（1グループ3～5名程度）の成果の発表会の予定です
- グループワークでは、研究室を複数グループ（人数によって変動）に分けますが、誰と組むことになるかはわかりませんので、みんなで助け合いレベルアップしましょう
 - ということで、まずはお互いの力量を把握するため、EP演習などで作ったものを紹介し合いましょう

今日やること



- 前回やった記録をどう再現する？
- ハイスコアをどう記録する？
- 実験などの記録をどう再現する？

どうやって再現する？



- 前回起動していたときの状態を引き継ぐには、どうしたらよいだろうか？

おきのどくですが
ほうけんのしょ1ばんは
きえてしまいました。

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！

ファイルを経由する



- プログラムとは別のファイルに一旦情報を記録しておき、そのファイルからまた情報を読み出す！
- セーブ&ロードしよう！
 - 記録する
 - 何かの値をファイルに記録しておく（ハイスコアや状態など）
 - 読み込む（再生する）
 - 記録した値をファイルから呼び出す（ハイスコアや状態など）

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("list.txt");  
String name = lines[0];  
int cost = int( lines[1] );
```

ファイルから読み込む



Windows

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

Mac OS

/ Users **/** nakamura **/** どこかのフォルダ **/** なんかのファイル

```
lines[0] = こんにちは みさなん おんげき ですか？  
lines[1] = わしたは げんき です。  
lines[2] = この ぶんよしう は いりぎす の ケブンツリジ  
lines[3] = だがいく の けゆきんう の けっか  
lines[4] = にんんげ は もじ を にしんき する とき その  
lines[5] = さしいよ と さいご の もさじえ あいてっれば  
lines[6] = じばんゆん は めくちちゃや でも ちんやと  
lines[7] = よめる という けゆきんう に もづいとて わざと  
lines[8] = もじの じんばゆん を いかれえて あまりす。  
lines[9] = どでうす？ ちんやと よやちめう でしょ？  
lines[10] =  
lines[11] = ちんやと よためら はゆしくを よしろく
```

上から順に読み込む

ファイルに書き込む



```
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
 - **str(何らかの値)** で整数などを文字列に変換

文字列保存演習



- 百人一首からひとつ歌をもってきて、プログラムで5,7,5,7,7を1行ずつuta.txtに保存するプログラムを作る
 - 5つの要素からなる文字列型の配列 `saveLines` を作成し、`uta.txt` に保存しましょう

<code>saveLines[0]</code>	花の色は
<code>saveLines[1]</code>	移りにけりな
<code>saveLines[2]</code>	いたづらに
<code>saveLines[3]</code>	わが身世にふる
<code>saveLines[4]</code>	ながめせし間に

文字列保存演習



```
// 5行分なので5つの要素からなる文字列配列を作る
```

```
String[] saveLines = new String[5];
```

```
// 1行ずつ値を放り込む
```

```
saveLines[0] =
```

```
saveLines[1] =
```

```
saveLines[2] =
```

```
saveLines[3] =
```

```
saveLines[4] =
```

```
// saveStringsでsaveLinesを1行ずつuta.txtに保存します
```

```
saveStrings("uta.txt", saveLines);
```

数値を保存する



- 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

クリック数を保存せよ



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、クリックした回数は随時保存し、次回起動した時にはそのクリック回数から増やしていくようにせよ

- クリックした回数を表示し続けるのであれば、クリック回数を保存する `click_count` という変数を定義してカウントアップしつつ表示する
- 終了 → 起動したときには当然0に戻ってしまうがどうやったらよいだろうか？

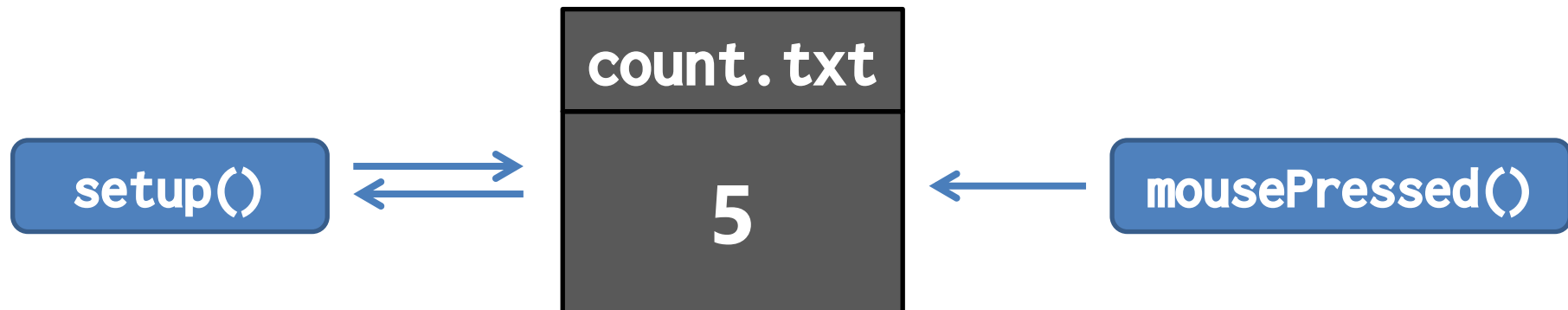
状態をファイルに保存する！

クリック数を保存せよ



クリックした回数は随時保存し、次回起動した時にはそのクリック回数から増やしていくようにせよ

- `mousePressed()` の度に `click_count` の値を増やし `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



```
int click_count = 0;

void setup() {
  size(600, 400);
  textSize(64);
  String[] lines = loadStrings("count.txt");
  if(lines != null ){
    click_count = int(lines[0]);
  }
}

void draw() {
  background(0, 0, 0);
  text(click_count, width/2, height/2);
}

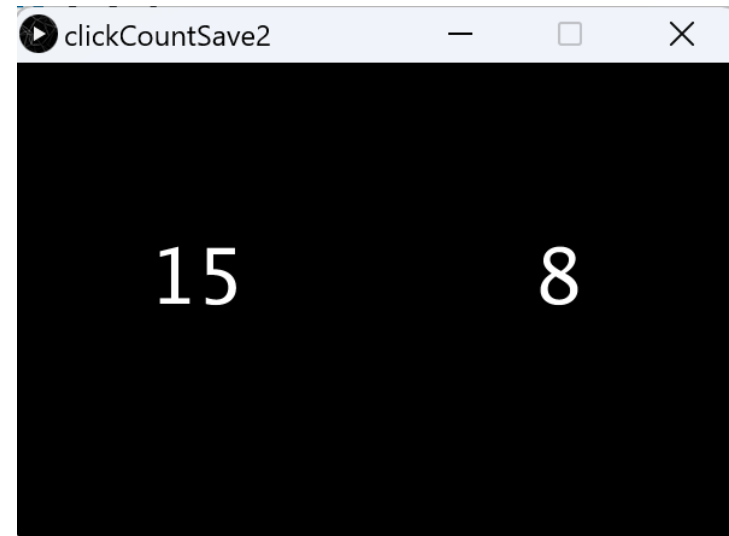
void mousePressed() {
  click_count++;
  String[] lines = new String[1];
  lines[0] = str(click_count);
  saveStrings("count.txt", lines);
}
```



左右のクリック数を保存



ウィンドウの左側・右側をそれぞれ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、クリックした回数は随時保存し、次回起動した時にはそのクリック回数から増やしていくようにせよ



左右のクリック数を保存



ウィンドウの左側・右側をそれぞれ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、クリックした回数は随時保存し、次回起動した時にはそのクリック回数から増やしていくようにせよ

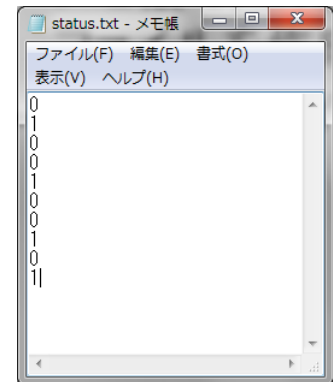
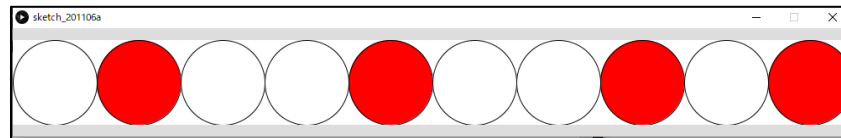
- `click_count1`, `click_count2` を用意しよう
- ファイルの1行目に `click_count1`、2行目に `click_count2` を保存しよう
- 読み込みも同じ！
 - `count.txt`を一度削除せよ！（エラーが出るので）

電光掲示板



1000x1000のウィンドウに横に10個，縦に1個並んだ電光掲示板について，丸をクリックする度に赤色，白色と塗りつぶしの色が入れ替わるようにせよ．また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





```
int[] lights = new int[10];
```

```
void setup()
```

```
{  
  size(1000, 100);  
  for(int x=0; x<10; x++){  
    lights[x] = 0;  
    x++;  
  }  
}
```

```
void draw()
```

```
{  
  background( 255 );  
  for(int x=0; x<10; x++){  
    if(lights[x] == 1){  
      fill(255, 0, 0);  
    } else {  
      fill(255);  
    }  
    ellipse(100*x+50, 50, 100, 100);  
  }  
}
```

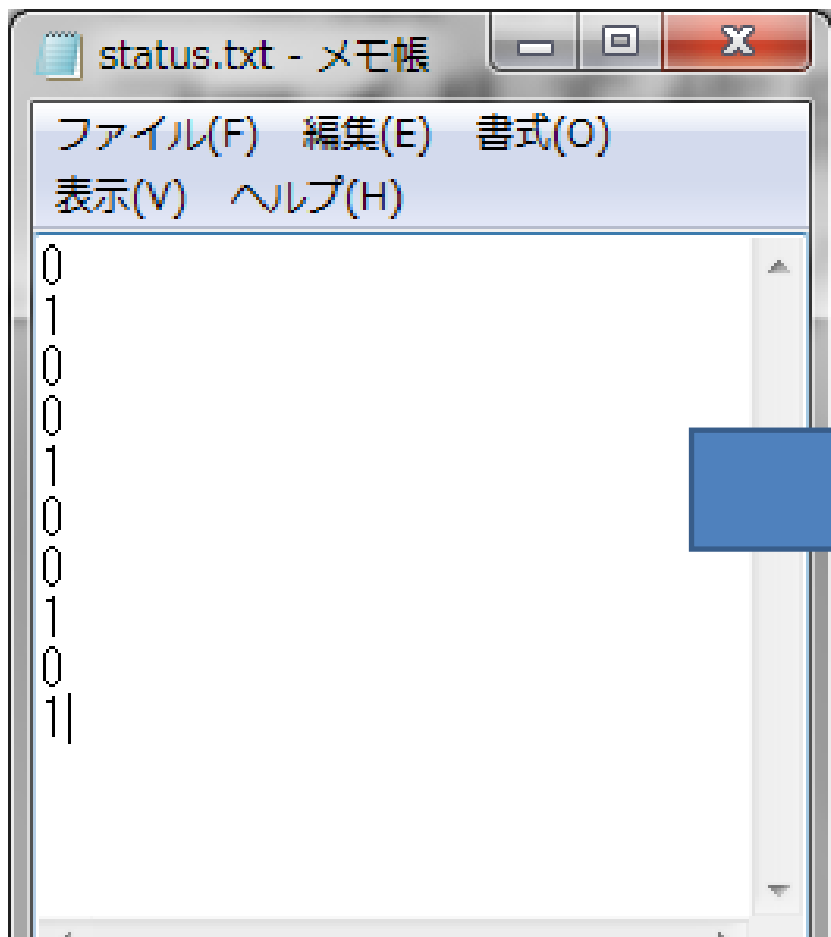
```
void mousePressed(){  
  for(int x=0; x<10; x++){  
    if(dist(100*x+50,50,mouseX,mouseY) <= 50){  
      if(lights[x] == 1){  
        lights[x] = 0;  
      } else {  
        lights[x] = 1;  
      }  
    }  
  }  
}
```

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



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.txt" );

  for(int x=0; x<10; x++){
    lights[x] = int( lines[x] );
  }
}
```

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

```
P lights_savveload | Processing 2.0.3
File Edit Sketch Tools Help
lights_savveload
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] );
Done Saving.
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.
5
```

電光掲示板プログラム



```
void mousePressed(){
    for(int x=0; x<10; x++){
        if(dist(100*x+50, 50, mouseX, mouseY) <= 50){
            if(lights[x] == 1){
                lights[x] = 0;
            } else {
                lights[x] = 1;
            }
        }
    }
}

String[] saveLines = new String[10];
for(int x=0; x<10; x++) {
    saveLines[x] = str( lights[x] );
}
saveStrings("status.txt", saveLines);
}
```

電光掲示板プログラム



```
void mousePressed(){
  for(int x=0; x<10; x++){
    if(dist(50+100*x, 50, mouseX, mouseY) <= 50){
      // 1,0,1,0と切り替えるならこの方法でもOK
      lights[x] = 1 - lights[x];
    }
  }
}

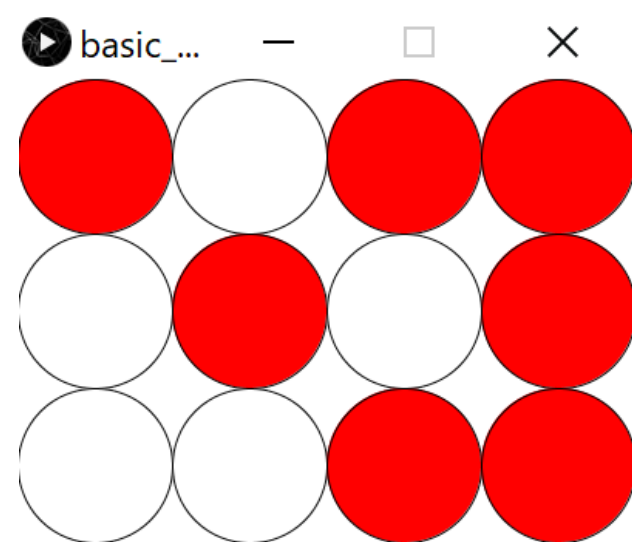
String[] saveLines = new String[10];
for(int x=0; x<10; x++) {
  saveLines[x] = str( lights[x] );
}
saveStrings("status.txt", saveLines);
}
```

電光掲示板（多次元）



電光掲示板のプログラムを改良して縦に3個、横に4個となるように変更し、左下のように記述したstatus.txt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。さらに、クリックのたびに保存し、次回の起動時に結果を引き継ぐようにせよ

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





```
int[][] lights = new int[4][3];
```

```
void setup() {  
  size(400, 300);  
  for(int x=0; x<4; x++){  
    for(int y=0; y<3; y++){  
      lights[x][y] = 0;  
    }  
  }  
}
```

```
void draw() {  
  background( 255 );  
  for(int x=0; x<4; x++) {  
    for(int y=0; y<3; y++) {  
      if(lights[x][y] == 1){  
        fill(255, 0, 0);  
      } else {  
        fill(255);  
      }  
      ellipse(x*100+50, y*100+50, 100, 100);  
    }  
  }  
}
```

```
void mousePressed() {  
  for(int x=0; x<4; x++){  
    for(int y=0; y<3; y++){  
      if(dist(x*100+50, y*100+50, mouseX, mouseY) <= 100){  
        lights[x][y] = 1 - lights[x][y];  
      }  
    }  
  }  
}
```

電光掲示板（多次元）



ヒント

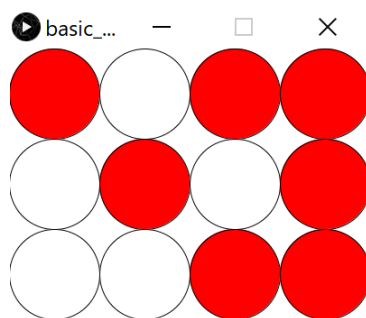
- `status.txt` というファイルをまず用意する（半角であることに注意！）
- `lights` という名前の 4×3 の配列を作って電光掲示板として実現
- 起動した際にファイル `status.txt` から1行ずつ取得し、そのそれぞれの値を配列に代入していく
 - 1行の値を読み込む時は、各値がカンマ区切りであることを考慮し、カンマごとに区切る場合は `split` を使う
- クリックされるたびに保存することにし、保存する際にも、 4×3 の配列の値をString型の配列に放り込んでいく

電光掲示板（多次元）



- 0行目: `lines[0]`は1,0,1,1という値
 - カンマで区切ったとき、1と0と1と1という値が続いていることになるが、この0番目、1番目、2番目がそれぞれ左上、中左上、中右上、右上に対応つまり0番目は`lights[0][0]`、1番目は…
- 1行目: `lines[1]`は0,1,0,1という値
- 2行目: `lines[2]`は0,0,1,1という値

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



<code>lights[0][0]</code>	<code>lights[1][0]</code>	<code>Lights[3][0]</code>
<code>lights[0][1]</code>	<code>lights[1][1]</code>	<code>lights[3][1]</code>
<code>lights[0][2]</code>	<code>lights[1][2]</code>	<code>lights[3][2]</code>

カンマ区切りの取得



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

```
// linesという配列に1行ずつ文字列を読み込む
String[] lines = loadStrings( "status.txt" );

// lines.lengthですべての行数を取得できる
for(int y=0; y<lines.length; y++ ){
    // line[i] ( i行目 ) の文字列をカンマで分割
    String[] data = split( lines[y], ',' );
    // 0番目、1番目、2番目の値を整数に変換して代入
    lights[ ? ][ ? ] = int( data[0] );
    lights[ ? ][ ? ] = int( data[1] );
    lights[ ? ][ ? ] = int( data[2] );
    lights[ ? ][ ? ] = int( data[3] );
}
```

splitで分割して読み込む



loadStringsしてカンマで分割して読み込み！

```
int[][] lights = new int[4][3];

void setup() {
  size(400, 300);

  // linesは3行入っている
  String[] lines = loadStrings("status.txt");
  for(int y=0; y<3; y++){
    // 「,」で分割すると4つに分けることができる
    String[] data = lines[y].split(",");

    // 整数に変換して代入しよう！
    lights[0][y] = int(data[0]);
    lights[1][y] = int(data[1]);
    lights[2][y] = int(data[2]);
    lights[3][y] = int(data[3]);
  }
}
```

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

splitで分割して読み込む



loadStringsしてカンマで分割して読み込み！

```
int[][] lights = new int[4][3];

void setup() {
  size(400, 300);

  // linesは3行入っている
  String[] lines = loadStrings("status.txt");
  for(int y=0; y<3; y++){
    // 「,」で分割すると4つに分けることができる
    String[] data = lines[y].split(",");
    for(int x=0; x<4; x++){
      // 整数に変換して代入しよう！
      lights[x][y] = int(data[x]);
    }
  }
}
```

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

カンマ区切りで保存！



str()で文字列に変換してカンマでつなぐ

```
// 3行分なので3つの要素からなる文字列配列を作る
String[] saveLines = new String[3]
// 1行ずつ値をカンマ区切りで代入していく
for(int y=0; y<lights.length; y++){
    // str()で文字列に変換して、「,」で4つの値をつなぐ
    // 4つだけなので直指定してますが、forで繰り返してもOK
    saveLines[ ? ] = str( ??? ) + ","
                    + str( ??? ) + ","
                    + str( ??? ) + ","
                    + str( ??? );
}
// saveStringsでsaveLinesを1行ずつstatus.txtに保存します
saveStrings("status.txt", saveLines);
```

strでくっつけて保存



```
void mousePressed() {
  for(int x=0; x<4; x++){
    for(int y=0; y<3; y++){
      if(dist(x*100+50,y*100+50,mouseX,mouseY)<=50){
        lights[x][y] = 1 - lights[x][y];
      }
    }
  }

  // 保存するのは3行
  String[] lines = new String[3];
  for(int y=0; y<3; y++){
    lines[y] = str(lights[0][y]) + "," + str(lights[1][y]) + "," + str(lights[2][y]);
  }
  saveStrings("status.txt", lines);
}
```

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

ネットからデータを読み込



- ProcessingのloadStringsは便利なことにネット上の情報を読み込むこともできる！
- こんな感じで書くだけ！ 超楽！

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

```
String[] lines = loadStrings( "http://nkmr.io/lecture/2020/scores.txt");
```



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

```
// 0から10なので11個の要素からなる配列を作る
```

```
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/2020/scores.txt" );
```

```
    for(int i=0; i<lines.length; i++){
```

```
        scores[ int( lines[i] ) ]++;
```

```
    }
```

```
    for(int i=0; i<scores.length; i++){
```

```
        println(i, scores[i]);
```

```
    }
```

```
}
```