

後ろ5列は着席禁止
3人掛けの中央は着席禁止

プログラミング入門 Processingプログラミング 第10回

九州産業大学 理工学部情報科学科
神屋郁子
(pp@is.kyusan-u.ac.jp)

時限	クラス
水1	機械 (クラス3)
水2	機械 (クラス1)
水4	電気 (B1、B2)

今後の予定

第10回：画像の表示と音の再生

第11回：応用課題プログラムの開発

第12回：Wordの基本操作と 応用課題プログラムについて
Wordを用いたレポート作成

第13回：Excelの基本操作と

Excel VBAによるプログラミング

第14,15回：Javaプログラミング
定期試験

2

宿題

- 次回は、応用課題の回
 - これまで作ったプログラムを振り返り、どんなプログラムを作るか、どんなアレンジをするか、予め考えてくること。
 - これまでの講義資料を次回持ってくること。

3

第10回の内容

- 画像、画像処理
 - 画像の読み込みと表示
 - 画像の加工
- 効果音、音楽
 - 効果音の読み込みと再生
 - 音楽の読み込みと再生
 - 音データの視覚化

応用課題の回の
アレンジに
役立てよう！

4

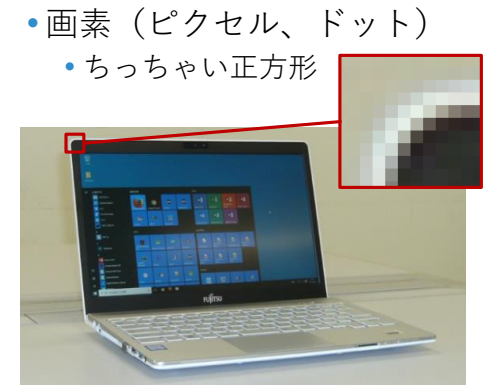
画像

- 画像の表示
 - これまでは図形を表示
 - 画像ファイルの読み込みと表示
 - 図形を動かす代わりに画像を動かせる！
 - 跳ね返るボールをキャラクターにしたり
 - ゲームでとんでくるボールやキャッチするバーをキャラクターにしたり
- 画像処理
 - 画像も数値データの集まり
 - 演算することで加工できる

5

画像とは

- 画像（静止画）
 - タテ・ヨコに切って画素に分解
 - 画素ごとの色合いを**数値化**して記憶
 - 右図を参照
- 画像ファイル
 - 拡張子 bmp, gif, png, jpg など



6

画像データ

- 色のデータは、RGB 8bit(0/1が8個) × 3
- 8bitは、16進数で 00からFFまで、10進数で 0から255まで
 - 例 FF 64 00 赤が強く緑を含む
R G B (オレンジ色)
- &を使うと値を取り出せる
 - 0xFF**FF**0000 だと R が取り出せる (c & 0xFFFF0000)
 - 0xFF00**FF**00 だと G が取り出せる (c & 0xFF00FF00)
 - 0xFF0000**FF** だと B が取り出せる (c & 0xFF0000FF)

7

効果音や音楽

- 効果音や音楽の再生
 - 音のファイルの読み込みと再生
 - これまでのプログラムに音を追加できる！
- 音データの種類
 - WAV 効果音
 - MP3 音楽
- 音の可視化
 - 音も数値データの集まり
 - 音の大きさ、周波数

8

画像、効果音、音楽

- 変数宣言（型 名前;）
`PImage img;`
`AudioSample audio;`
`AudioPlayer music;`
- 読み込み（代入）
`img = loadImage("?????.jpg");`
`audio = minim.loadSample("?????.wav", 2048);`
`music = minim.loadFile("?????.mp3", 2048);`
- 描画、再生
`image(img, 0, 0);`
`audio.trigger();`
`music.play();`

9

作業

- 著作権フリーの画像を用意
 - 例 フォトスク <http://photosku.com/>
- 画像の表示
- 効果音の再生
- 音楽の再生
- 応用

10

注意点

- 画像や音のファイルはフリー素材や著作権上問題ないものを使用するように（もしくはレポート用と自分用に2つ作る）
- 今回は、画像や音のファイルを使うので、プログラムの保存場所がわからなくなるのを避けるために、プログラムを入力したりファイルを追加する前に一度プログラムを保存しておくのをお勧めします。

11

[プログラムの作成手順]

演習1

画像の表示

- 著作権フリーの画像を2枚保存する
- [スケッチ]メニューの[ファイルを追加]をクリックし画像ファイルを読み込む
- [スケッチ]メニューの[スケッチフォルダーを開く]をクリックしてdataフォルダーの中にそのファイルがあることを確認する
- プログラムを記述する

12

[プログラムの作成手順]

演習1


• 著作権フリーの画像を2枚保存する

画像の表示

- [スケッチ]メニューの[ファイルを追加]をクリックし画像ファイルを読み込む
- [スケッチ]メニューの[スケッチフォルダーを開く]をクリックしてdataフォルダーの中にそのファイルがあることを確認する
- プログラムを記述する

13

[画像の保存(はじめに)]

- ブラウザを起動して、フォトスク (<http://photosku.com/>) のページへ移動
 - この後説明用に使用するブラウザはGoogle Chrome () なので、Google Chromeを利用してフォトスクのページにアクセスすること。
 - 自分の撮った写真などを使用しても良いが、芸能人やアニメのキャラクターなどの画像を使用したり、Twitterなどにある、人が撮った/描いた写真や絵(俗に言う"拾い画")は使用しないこと
 - 他の画像を探したい人は「画像 素材 フリー」のようなキーワードで検索すると良い。

14

[画像の保存(1/2)]

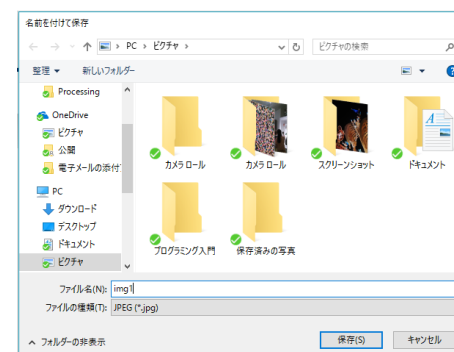
- トップページ左側のカテゴリーをクリックし、出てきた写真から使用する写真を選択し、写真を右クリックして「名前をつけて画像を保存」をクリック



15

[画像の保存(2/2)]

- ファイル名のところに記入されたファイル名を消してファイル名を入力



保存場所：ピクチャ
1つ目のファイル：img1
2つ目のファイル：img2

※これ以外の場所、ファイル名でも良いが、これ以降の説明で使うので、よくわからない人はこの場所・この名前にしておくこと。

16

[プログラムの作成手順]

演習1

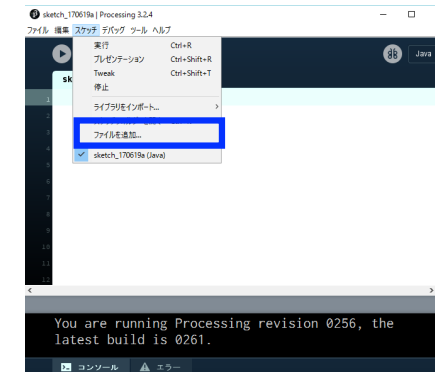
- 著作権フリーの画像を2枚保存する
- **[スケッチ]メニューの[ファイルを追加]をクリックし画像ファイルを読み込む**
- [スケッチ]メニューの[スケッチフォルダーを開く]をクリックしてdataフォルダーの中にそのファイルがあることを確認する
- プログラムを記述する

画像の表示

17

[画像ファイルの読み込み(1/2)]

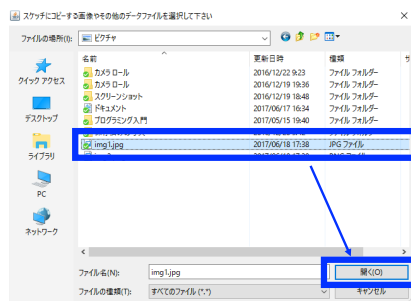
- **[スケッチ]メニューの[ファイルを追加]をクリック**



18

[画像ファイルの読み込み(2/2)]

- 保存した画像ファイル（ピクチャフォルダが表示されていない場合は[ライブラリ]から[ピクチャ]を選択）をクリックし、「開く」をクリック



19

[プログラムの作成手順]

演習1

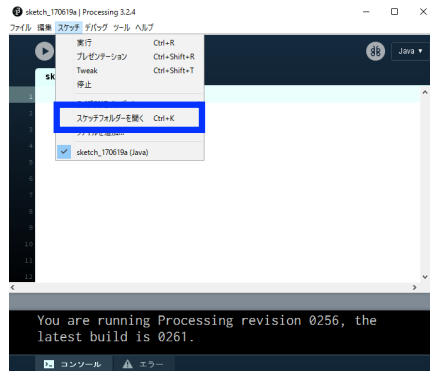
- 著作権フリーの画像を2枚保存する
- **[スケッチ]メニューの[ファイルを追加]をクリックし画像ファイルを読み込む**
- **[スケッチ]メニューの[スケッチフォルダーを開く]をクリックしてdataフォルダーの中にそのファイルがあることを確認する**
- プログラムを記述する

画像の表示

20

[画像の確認 (1/3)]

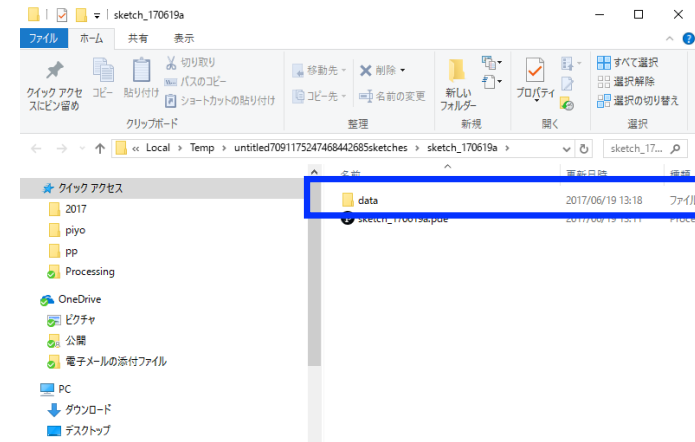
- [スケッチ] メニューの[スケッチフォルダーを開く]をクリック



21

[画像の確認 (2/3)]

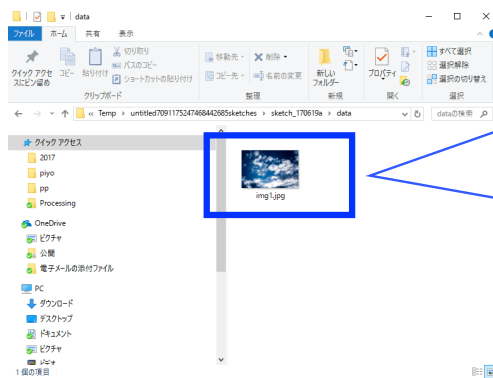
- [data] フォルダをクリック



22

[画像の確認 (3/3)]

- 保存したファイルがあるか確認



ファイル名の後ろには拡張子 (スライドの例では.jpg) がついており、拡張子はファイルによって様々。プログラムに記述するので、今回dataフォルダに置いたファイル名は「ファイル名+拡張子」の形でどこかにメモしておくこと。

23

[プログラムの作成手順]

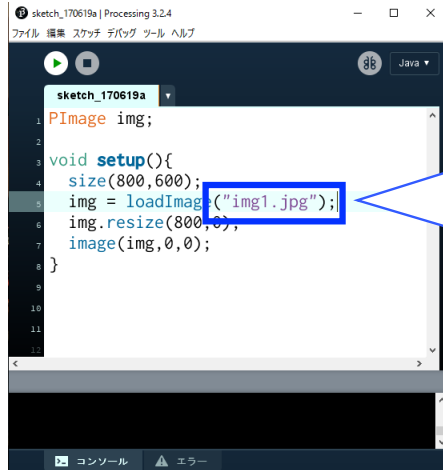
演習1

画像の表示

- 著作権フリーの画像を2枚保存する
- [スケッチ]メニューの[ファイルを追加]をクリックし画像ファイルを読み込む
- [スケッチ] メニューの[スケッチフォルダーを開く]をクリックしてdataフォルダーの中にそのファイルがあることを確認する
- プログラムを記述する

24

[プログラムの記述]



22ページで確認した**ファイル名.拡張子**を記入。ファイル名の
大文字や小文字は間違えない
ように書くこと。

25

```
PImage img;
```

```
void setup(){  
  size(800, 800);  
  img = loadImage("img1.jpg");  
  img.resize(800,0);  
  image(img,0,0);  
}
```

loadImageの命令で
画像ファイルを読み込む。

resizeの命令で拡大縮小。
横800ドットに合わせる。

imageの命令で
読み込んだ画像を表示。

演習1

画像の表示

26

[プログラムの作成手順]

効果音の再生

演習2

- [スケッチ]メニューの[ファイルを追加]をクリックし、WAV形式のファイルを読み込む。
- [スケッチ]メニューの[スケッチフォルダーを開く]をクリックして、dataフォルダーの中にファイルがあることを確認する。
- プログラムを入力する前に[スケッチ]メニューの[ライブラリーをインポート]からMinimを選ぶ

27

[プログラムの作成手順]

効果音の再生

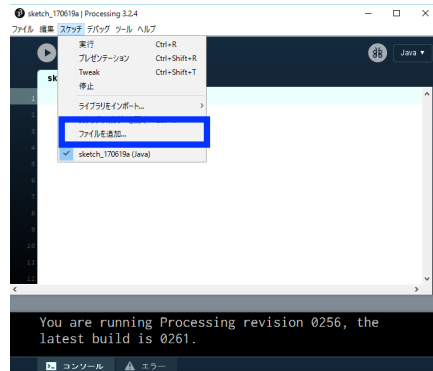
演習2

- **[スケッチ]メニューの[ファイルを追加]をクリックし、WAV形式のファイルを読み込む。**
- [スケッチ]メニューの[スケッチフォルダーを開く]をクリックして、dataフォルダーの中にファイルがあることを確認する。
- プログラムを入力する前に[スケッチ]メニューの[ライブラリーをインポート]からMinimを選ぶ

28

[音データの読み込み(1/2)]

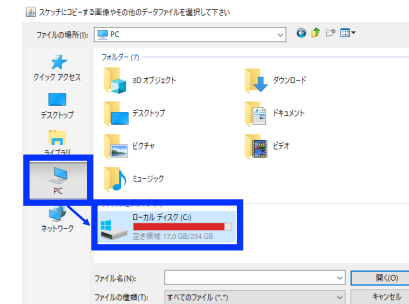
- [スケッチ]メニューの[ファイルを追加]をクリック



29

[音データの読み込み(2/2)]

- 例) C:¥Windows¥mediaのtada.wav を開く
 - cドライブの**Windowsフォルダ**の**mediaフォルダ**にあるtada.wavというファイルを開く



30

[プログラムの作成手順]

効果音の再生

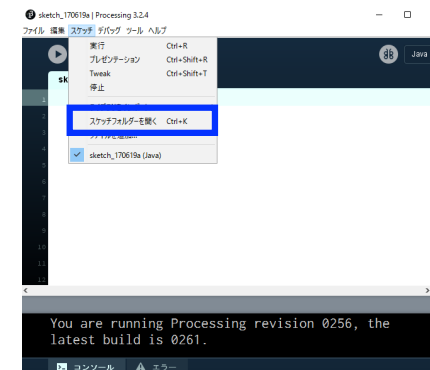
演習6

- [スケッチ]メニューの[ファイルを追加]をクリックし、WAV形式のファイルを読み込む。
- [スケッチ]メニューの[スケッチフォルダーを開く]をクリックして、**data**フォルダの中にファイルがあることを確認する。
- プログラムを入力する前に[スケッチ]メニューの[ライブラリーをインポート]からMinimを選ぶ

31

[音データの確認 (1/3)]

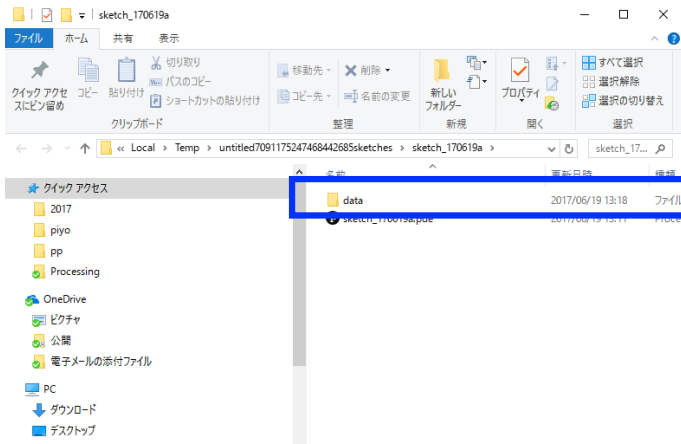
- [スケッチ]メニューの[スケッチフォルダーを開く]をクリック



32

[音データの確認 (2/3)]

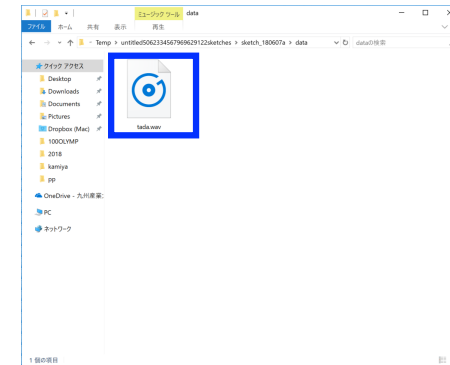
- [data] フォルダをクリック



33

[音データの確認 (3/3)]

- ファイルがあるか確認



34

[プログラムの作成手順]

効果音の再生

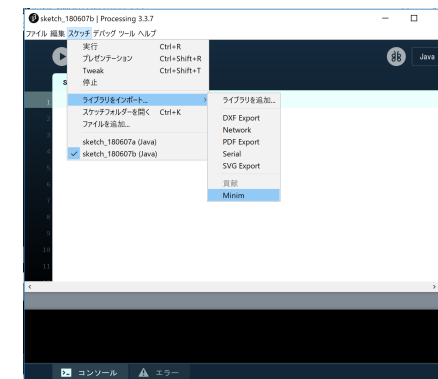
演習2

- [スケッチ]メニューの[ファイルを追加]をクリックし、WAV形式のファイルを読み込む。
- [スケッチ]メニューの[スケッチフォルダーを開く]をクリックして、dataフォルダの中にファイルがあることを確認する。
- プログラムを入力する前に[スケッチ]メニューの[ライブラリーをインポート]からMinimを選ぶ

35

[ライブラリーのインポート (1/2)]

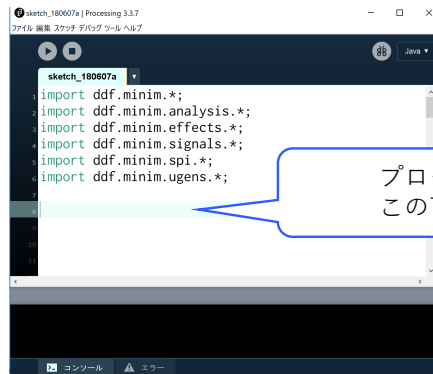
- [スケッチ]メニューの[ライブラリーをインポート]からMinimを選択



36

[ライブラリのインポート (2/2)]

- import…の行が追加される



プログラムは
この下に記述

37

```
Minim minim;  
AudioSample audio;
```

演習2

効果音の再生

```
void setup(){  
    minim = new Minim(this);  
    audio = minim.loadSample("tada.wav",2048);  
}  
void draw(){  
}  
void mousePressed(){  
    audio.trigger();  
    ellipse(mouseX,mouseY,50,50);  
}
```

loadSampleの命令で
音のファイルを読み込む

triggerの命令で音を再生
する

38

```
Minim minim;  
AudioPlayer music;  
String s;
```

演習3

音楽の再生

```
void setup(){  
    size(400,400);  
    minim = new Minim(this);  
    s = "http://www.is.kyusan-u.ac.jp/~goshi/d/irish.mp3";  
    music = minim.loadFile(s,2048);  
    music.play();  
}  
void draw(){  
}
```

loadFileの命令で
音楽ファイルを読み込む

playの命令で
音楽を再生

39

プログラムの提出 (演習点)

- 画像、効果音、音楽どれかのプログラムを提出
 - なるべくたくさんアレンジ/改良したものを提出すること
 - 画像や音のファイルはフリー素材や著作権上問題ないものを使用するように
- K's Lifeのレポート機能から
- 締め切りは6日後の23:59
 - それ以降も受け取るが**減点**する
 - もっと改良したくてもこの時間に一度提出しよう
 - (削除して提出しなおし可能)
- 評価はK's Life上ではしない

40

提出する内容

- 以下をコメントとして入力
 - 今回の内容の概要（画像・効果音・音楽のどれを選んだか）
 - 工夫した点
 - 質問（何かあれば）・感想（簡単だった・難しかったなど）
 - 入らない場合は3つ目の提出ファイルとして追加してもよい
- 提出ファイルとして以下の2つ
 - プログラム(*.pde)
 - 実行画像(PNGまたはJPG)
- 提出ファイルの名称にはそれぞれ学籍番号を入力

41

```
void draw(){
```

```
background(0);
```

```
float left = 0;
```

```
float right = 0;
```

```
for(int i = 0; i < music.left.size(); i++){
```

```
    left += abs(music.left.get(i));
```

```
    right += abs(music.right.get(i));
```

```
}
```

```
left = 1000*left/music.left.size();
```

```
right = 1000*right/music.right.size();
```

```
noStroke();
```

```
ellipse(width/3, height/2, left, left);
```

```
ellipse(width*2/3, height/2, right, right);
```

```
}
```

左右の音量用の変数を宣言

補足a

音楽と図形の連動

for文の繰り返しで音量データを作る

丸の大きさは音量に応じて数値1000を調整する

左右の音量に応じた大きさの丸を描く

42

```
Minim minim;
```

```
AudioPlayer music;
```

```
String s;
```

```
AudioMetaData Data;
```

```
void setup(){
```

```
    size(400,400);
```

```
    minim = new Minim(this);
```

```
    s = "http://www.is.kyusan-u.ac.jp/~goshi/d/irish.mp3";
```

```
    music = minim.loadFile(s,2048);
```

```
    music.play();
```

```
meta = music.getMetaData();
```

```
}
```

補足b

MP3メタ情報の表示

43

```
void draw(){
```

```
    ...
```

```
    noStroke();
```

```
    ellipse(width/3, height/2, left, left);
```

```
    ellipse(width*2/3, height/2, right, right);
```

```
text("Album: " + meta.album(),10,20);
```

```
text("Artist: " + meta.author(),10,40);
```

```
text("Title: " + meta.title(),10,60);
```

```
text("Comment: " + meta.comment(),10,80);
```

```
}
```

補足b

MP3メタ情報の表示

44

```

Minim minim;
AudioPlayer music;
String s;
AudioMetaData Data;

```

FFT fft;

```

void setup(){
  size(400,400);
  minim = new Minim(this);
  s = "http://www.is.kyusan-u.ac.jp/~goshi/d/irish.mp3";
  music = minim.loadFile(s,2048);
  music.play();
  meta = music.getMetaData();

  fft = new FFT( music.bufferSize(), music.sampleRate());
}

```

補足c

周波数スペクトルの表示

45

```

void draw(){
  ...
  text("Album: " + meta.album(),10,20);
  text("Artist: " + meta.author(),10,40);
  text("Title: " + meta.title(),10,60);
  text("Comment: " + meta.comment(),10,80);

```

```

stroke(255);
fft.forward(music.mix);
for(int i = 0; i < width/4; i++){
  line(i*4, height, i*4, height - fft.getBand(i)*4);
}
}

```

補足c

周波数スペクトルの表示

46

```

Pimage img;

void setup(){
  size(800, 800);
  img = loadImage("img1.jpg");
  img.resize(800,0);
  image(img,0,0);
}

```

```

void draw(){
}

void mousePressed(){
  color col = img.get(mouseX,mouseY);
  println(mouseX+", "+mouseY+" "+hex(col,6));
}

```

getの命令で
色の値を取得する。

mousePressedメソッドは
マウスのボタンが
押された時に実行される処理

補足d

マウスでクリックした
場所の色の値を表示

47

```

void mousePressed(){
  color col = img.get(mouseX,mouseY);
  println(mouseX+", "+mouseY+" "+hex(col,6));
  for(int y = 0; y < 100; y++){
    for(int x = 0; x < 100; x++){
      img.set(mouseX+x,mouseY+y,(c & 0xFFFF0000));
    }
  }
  image(img,0,0);
}


```

setの命令で
色を変更する。

補足e

マウスでクリックした
場所の付近の色を
赤くする。

48



```
PImage img;
```

補足f

```
void setup(){  
  size(800,800);  
  img = loadImage("img1.jpg");  
  img.resize(800,0);  
  image(img,0,0);  
}
```

もう1つ別の
画像を表示

```
void draw(){  
}
```