

1. 画像の表示

まず、著作権フリーの画像を用意する。例 フォトスク <http://photosku.com/>
[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。プログラム中でその画像の **ファイル名** を使うこと。ここではファイル名は例として **img1.jpg** を使う。

```
PImage img;  
void setup(){  
  size(800, 600);  
  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
2つ目のデータが0の場合は
縦横比が維持される。
0より大きい場合は、
縦の大きさの指定

image の命令で
読み込んだ画像を表示

2. 効果音(WAV)の再生

新たにウインドウを開く。([ファイル]メニューから[新規])
[スケッチ]メニューの[ファイルを追加]をクリックしWAV形式の音データのファイルを読み込む。ここでは例として、PC→Cドライブ→Windowsフォルダー→Mediaフォルダにある **tada.wav** を使用する(C:\Windows\Media\tada.wav)。[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[ライブラリをインポート]からから Minim を選べと import ... の行が追加される。import ... の行の続きに、以下を入力する。実行し、クリックすると音が再生される。

```
Minim minim;  
AudioSample audio;  
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

3. 音楽(MP3)の再生

新たにウインドウを開く。([ファイル]メニューから[新規])
インターネット上のMP3のファイルを保存せずにそのまま再生することができるので、以下では、

<http://www.is.kyusan-u.ac.jp/~goshi/d/irish.mp3> を使用する。
もしくは、mp3形式の音楽データを1.や6.と同じ要領で置く。
[スケッチ]メニューの[ライブラリをインポート]からから Minim を選ぶ。以下を入力し、実行すると音楽が再生される。



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

```
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() {  
}
```

今回、音源データ(Reels)をこの講義のために提供してくれたのは、tricolor, John John Festival, 0' Jizo 等のアイルランド音楽のバンドメンバーとして活躍されている中村大史さんです。東京在住ですが来福されることも多く、福岡で演奏を聞く機会もたびたびあります。この場を借りてお礼申し上げます。ありがとうございます。
<http://hirofuminakamura.com/>
福岡でも、アイルランド音楽のセッション、ライブ、練習会など、様々な活動が行われています。
(興味のある人は合志まで)

play の命令で
音楽を再生する。

loadFile の命令で
音楽のファイルを読み込む
～は、SHIFT キーを押しながら
ひらがなの「へ」のキー

4. 応用

画像表示と効果音再生や音楽再生を組み合わせよう。

以前作成したゲームにBGMや効果音を追加することもできる。

画像をPNG形式の透過画像を使うと背景との重ね合わせもできる。

透過画像を作るには例えば 手軽に透明png <http://www.officedaytime.com/toumei/>

今までの丸や四角を動かすプログラムの丸や四角を画像で置き換えることも可能。

興味のある人は裏ページの補足も試してみよう。

プログラムの提出(10週) K's Lifeのレポート機能にて提出

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

提出するプログラムは、画像、効果音、音楽どれでもよい(なるべくアレンジをたくさんしたものを選んで提出すること。)

- ・以下をコメントとして入力(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)
今回の内容の概要、工夫した点、質問(何かあれば)・感想
- ・提出ファイルとして 画像か音かどちらかのプログラム(*.pde)と実行画像(PNGまたはJPG)

補足 興味のある人は以下も試してみよう。

a. 音楽と図形の連動

3. で音の大きさと図形を連動させてみよう。music.left.get(i)で左の音の大きさを取得する。音も数値データなので丸の大きさに利用できる。以下を draw の中に追加する。

```
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);
```

左右の音量用に変数を宣言

for 文の繰り返しで音量データをつくる

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

b. MP3 のメタ情報の表示

次の変数を追加する。

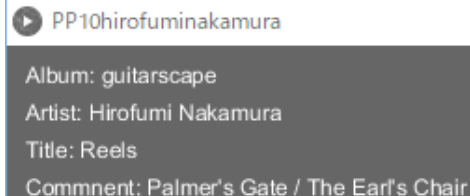
```
AudioMetaData meta;
```

setup の最後に次の文を追加する。

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

以下を draw の中に追加する。

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



c. 周波数スペクトルの表示

次の変数を追加する。

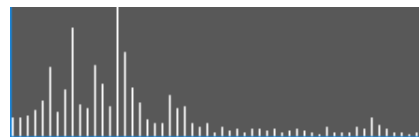
```
FFT fft;
```

setup の最後に次の文を追加する。

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

以下を draw の中に追加する。

```
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);
}
```



for 文の繰り返しで周波数毎に縦の線を表示

高さの調整は4の数値を変更

d. マウスでクリックした場所の色の値を表示する。

新たにウィンドウを開いて入力するか1. のプログラムを以下のように修正する。

img.get(x, y) で x, y 座標の画素の色のデータを取得する。

```
PImage img;
void setup(){
  size(800, 600);
  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));
}
```

mousePressed メソッド以外には、mouseReleased (ボタンを離した) や mouseMoved (移動した) 等がある。

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

get の命令で色の値を取得する。

e. マウスでクリックした場所の付近の色を赤くする。

以下を mousePressed メソッドの最後に追加する。img.set(x, y, c) で x, y 座標の画素の色を変更する。

```
for (int y = 0; y < 100; y++) {
  for (int x = 0; x < 100; x++) {
    color c = img.get(mouseX+x, mouseY+y);
    img.set(mouseX+x, mouseY+y, (c & 0xFFFF0000));
  }
}
image(img, 0, 0);
```

set の命令で色を変更する。

f. もう一つ別の画像を表示

d. に次の変数を追加する。

```
PImage img2;
```

setup の最後に次の文を追加する。

```
img2 = loadImage("?????");
```

```
img2.resize(100, 0);
```

画像を表示したい処理 (例えば draw のメソッド) の中に以下を追加する。

```
image(img2, x座標, y座標);
```

適当な大きさにサイズ調整 (resize)