プログラミング基礎

画像の読み込みとアニメーション

今回の内容

- 画像の読み込みと表示
- 変数の有効範囲
- ・アニメーションの基礎

画像の読み込みと表示

- プログラムで使いたい画像ファイルを用意 (gif, jpg, png)
- 2. プログラムに名前を付けて保存する
- 3. メニューから Sketch > Add File... から用意し た画像を追加する
- 4. loadImage 関数でプログラムに読み込む
 - Pimage 型の変数に入れる
- 5. image 関数で描画する

画像の読み込みと表示

(先に前のスライドの1~3をしておくこと)



変数の有効範囲

```
void setup() {
 size(400, 400);
 PImage img = loadImage("cat1-a.gif");
void draw() {
 background (255);
 image(img, mouseX, mouseY);
  image(img, mouseX, mouseY, img.width / 2, img.height / 2);
         setup の中で変数 img を作ると
         draw では使えない
Cannot find anything named "img"
```

変数の有効範囲(=scope)

• 有効範囲は {...} で区切られる

作るときのルール

1つの有効範囲内では、同じ名前のものを1個作れる

使うときのルール 内から外へ探し、最初に見つかったものを指す

image関数

- image(img, x, y)
- image(img, x, y, width, height)
- img – 描画したい PImage
- x, y – 描画する場所(左上の点)
- width, height
 - 描画する大きさ
 - 省略した場合は元画像の大きさになる

アニメーションの基礎

- ちょっとずつ違う絵を高速に繰り返し描画する
 パラパラマンガをさらに高速にしたような感じ
- Processing でアニメーション

 draw 関数を使う(繰り返し実行される)

まずは、複数の画像を用意して順番に描画していくプログラムを書いてみましょう

パラパラマンガ的アニメーション

String[] filenames = {"cat1-a.gif", "cat1-b.gif"};
PImage[] images;

```
void setup(){
```

size(95, 111); // ウィンドウのサイズを画像に合わせている
frameRate(2); // 1秒間にdrawを実行する回数を設定

```
images = new PImage[filenames.length];
for(int i = 0; i < filenames.length; i = i + 1){
    images[i] = loadImage(filenames[i]);</pre>
```

```
void draw(){
    background(255);
    int frame = frameCount % images.length;
    image(images[frame], 0, 0);
```

frameCount = draw 関数が実行された回数

課題(3まで必須)

- 1. 3枚以上の画像を用意してパラパラさせよう
- 2. パラパラする速度を調整しよう
 - 方法1: frameRate 関数を使う
 - 方法2:↓の部分を変える

int frame = frameCount % images.length;

3. さらにマウスに付いてくるようにしよう

- 注:方法1だと滑らかに付いてこない

提出方法

- 今までと違うので注意!
- ・ プロジェクトのフォルダを zip 圧縮する
 - 1. プロジェクトを保存
 - 2. Tools > Archive Sketch
 - 3. (名前).zip という名前のファイルが出来る
- 上記手順で作成した zip ファイルを 課題提出システムから提出