Processingによる画像処理プログラミング

Processing
Version3.3 Windows

電気電子システム学科 画像工学

1. Processing

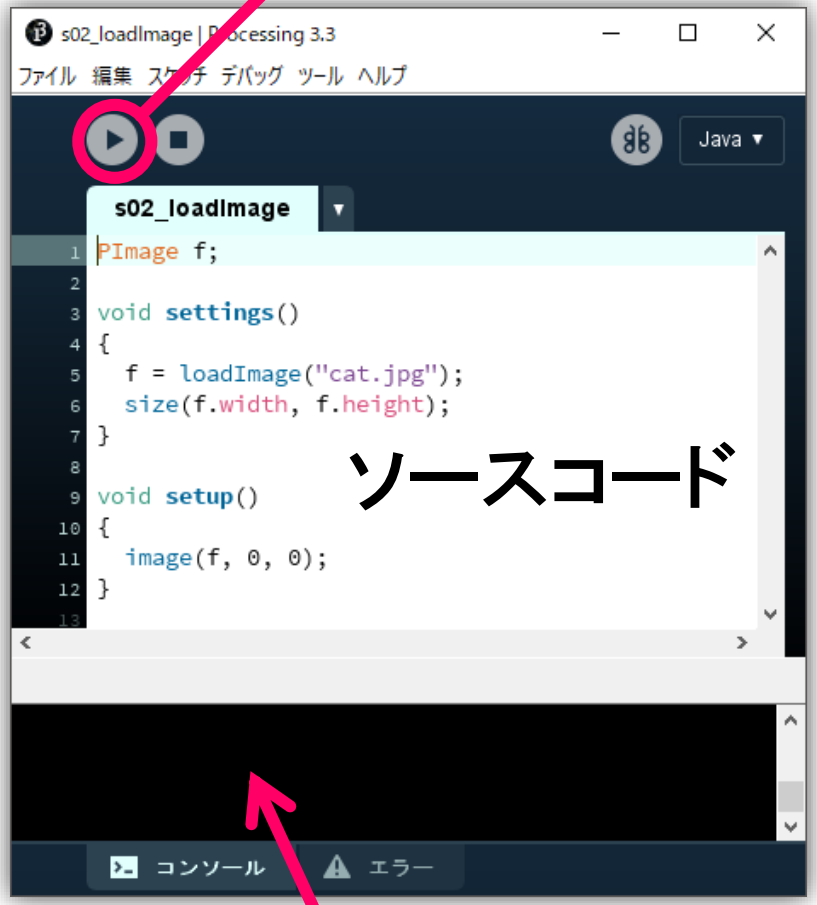
- Javaベースのプログラミング言語
- オープンソースの開発統合環境
- Windows、Mac OS X、Linuxで動作
- グラフィック描画（2D、3D）が簡単にできる

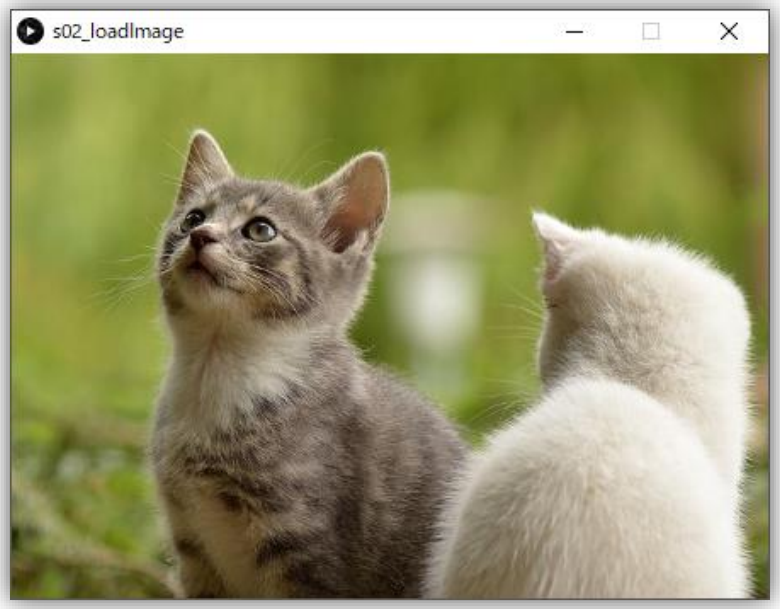
https://www.processing.org

2. 開発環境と実行画面

実行ボタン

実行画面





ソースコード

コンソール出力

3. プログラムの作成

① ソースコードを入力

② [ファイル]→[名前をつけて保存] で、保存場所と名前を指定

- 保存場所に指定した名前のフォルダが作られて、その中にソースファイルが保存される。
- プログラムはフォルダ単位で管理される。
- ソースファイル名とフォルダ名は同じ名前にしておかなければならない。

4. 画像ファイルの登録

① 原画像として使用する画像ファイル（jpg、pngなど）を用意

- 画像は原寸で表示されるので、高解像度の画像はあらかじめ縮小しておくと良い。

② [スケッチ]→[ファイルを追加]で、画像ファイルを指定

- プログラムの保存場所にdataフォルダが作られ、その中に画像ファイルがコピーされる。

5. プログラムの形式

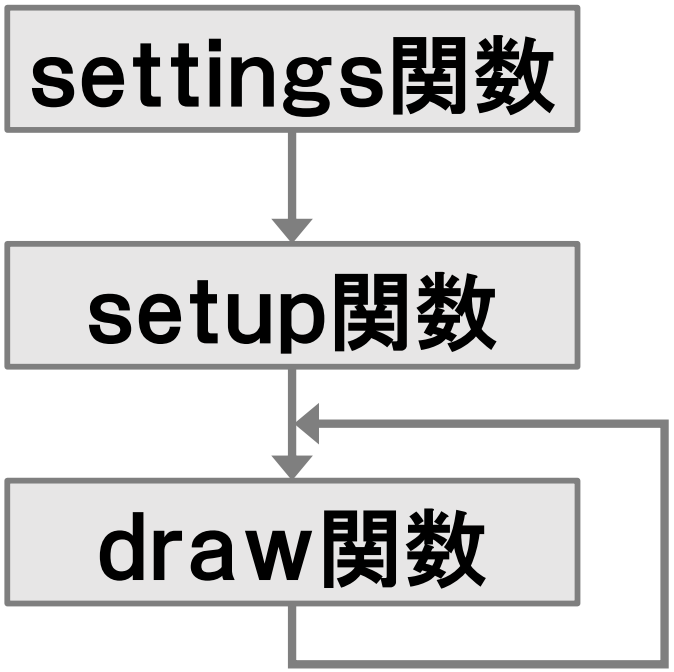
プログラムの基本形

```
void settings() {
  ....
}

void setup() {
  ....
}

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

実行順序



C言語とは異なる。（C言語は、main関数から開始する。）

※ 演習のプログラムでは、draw関数は使用していない。

6. 実行画面

実行画面のサイズ指定

書式　size(横幅, 縦幅);

- 横幅と縦幅を定数で指定する(変数は不可)。ただし、settings関数内に限り、変数でも指定できる。
- 基本的に、原画像のサイズと同じにすればよい。複数の画像を並べて表示するときは、横幅または縦幅を広げる。

7. コンソール出力

println(”文字列”);
文字列を表示する。

println(式);
式の値(変数の値)を表示する。

println(式, 式);
式の値を横に並べて表示する。

println(”文字列”, 式);
文字列と式の値を横に並べて表示する。

8. 変数型

	Processing	C言語(参考)
整数	int	int
実数	float	float
		double
色	color	

配列の宣言

C言語　int a[10];

Processing　int[ ] a = new int [10];

9. データ型変換

int変数に実数は代入できない。

```
int   a = 1;
float  x = 2.5;
a = x;
x = a;
```

×不可（実行できない）

○可能

型変換関数

int(値)

値を整数（小数点以下切り捨て）に変換する。

```
a = int( x );
```

○可能

10. 制御文・関係演算子

- if－else
- switch－case－break
- while
- do－while
- for
- ==、!=、<=、>=、&&、||

C言語と同じ書き方をする。

11. 画像データ型

PImage型（クラス）

書式　PImage　オブジェクト名（変数名）；

PImage型オブジェクト 1つにつき、画像1枚を格納できる。

画像（無地）の作成

PImage img；

img＝createImage（横幅，縦幅，RGB）；

横幅×縦幅のカラー画像をimgに作成する。

loadImageでファイルから読み込む場合は、前もって作成は不要。

12. 画像の入出力

PImage型

img

実行画面

img＝loadImage(″ファイル名″);

ファイル読み込み

画像ファイル

ファイル保存

img.save(″ファイル名″);

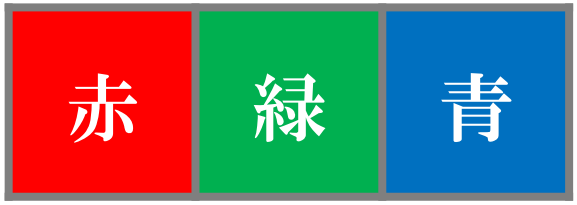
画面表示

image(img, x, y);

実行画面の座標(x,y)に画像の原点が位置するように表示される。

13. 色の表現

color型



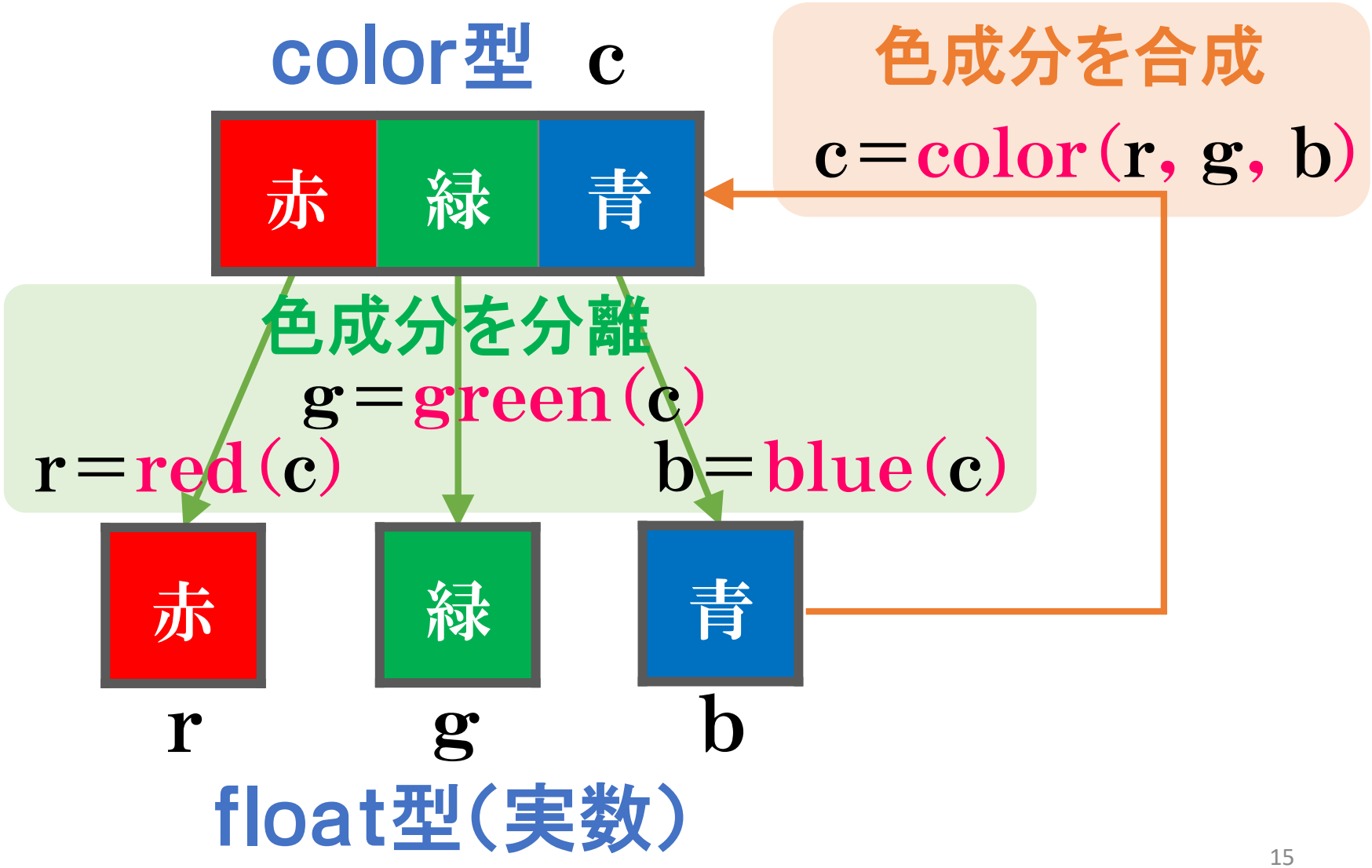
書式　color　変数名;

カラー画像の画素値（赤成分・緑成分・青成分：0～255の整数値）を表す。

赤、緑、青の各成分に分けるには、red関数、green関数、blue関数を使う。

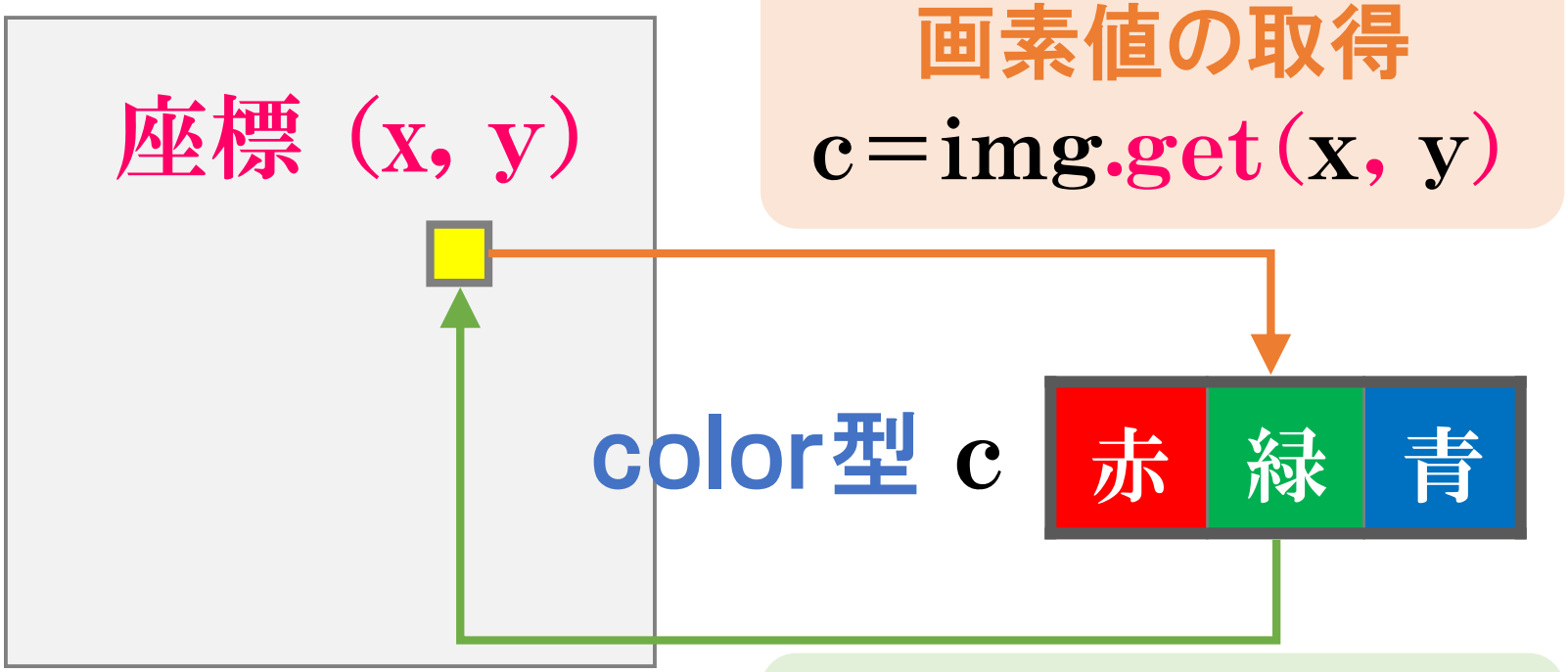
赤、緑、青の各成分から、color型の値を作るには、color関数を使う。

14. 色(画素値)の変換


color型 c
色成分を合成
c＝color(r, g, b)
赤
緑
青
色成分を分離
g＝green(c)
r＝red(c)
b＝blue(c)
赤
緑
青
r
g
b
float型(実数)

15. 画素値の読み書き

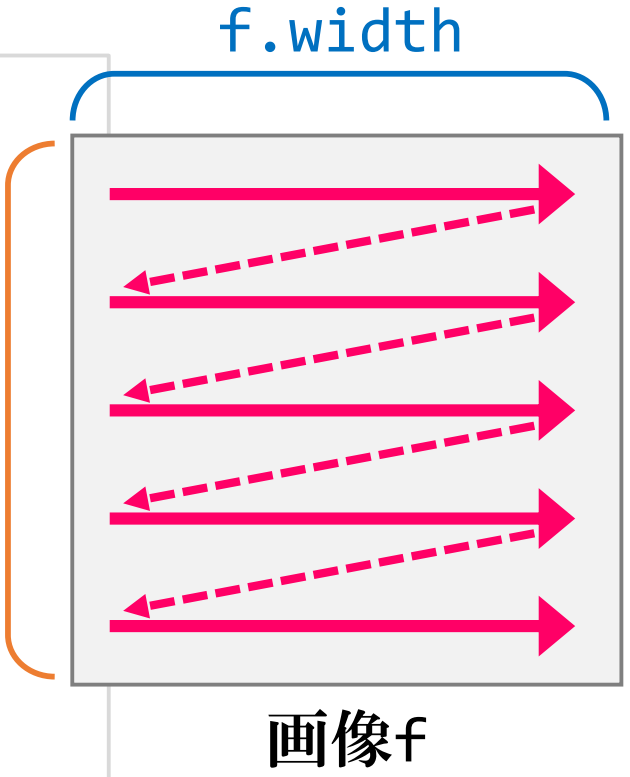
PImage型　img



画素値はcolor型で
表される。

16. ラスター走査

```
Pimage f;
int x, y;
float r, g, b;
(画像の読み込みなどの処理)
for(y=0; y<f.height; y++){
  for(x=0; x<f.width; x++){
    r =   red( f.get(x, y) );
    g = green( f.get(x, y) );
    b =  blue( f.get(x, y) );
    (画素値計算の処理)
    f.set(x, y, color(r, g, b));
  }
}
```



17. グレイスケール画像、2値画像

filter関数

PImage img;

img.filter(GRAY);

画像imgをグレイスケール画像に変換する。

1画素の赤成分、緑成分、青成分は同じ値になる。

img.filter(THRESHOLD, 閾値);

画像imgを2値化する。0か255の2値になる。

閾値は0～1の実数で与える(1は画素値255に相当する)。