Processingの使い方

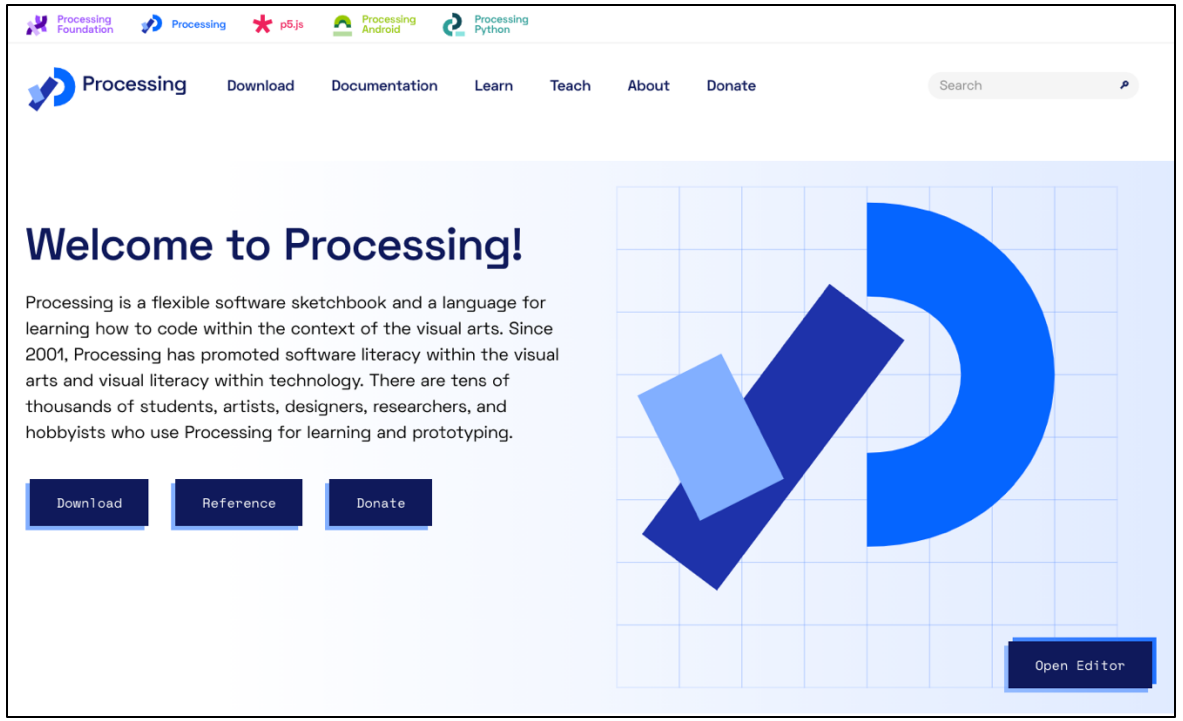
有本泰子

本資料のねらい

- 簡単な操作方法を理解し，Processingのプログラムを書いて実行することができる
- 動的なプログラムを書くための基本的な関数の機能について理解する
- Processingにおける動的メモリ確保について理解し，利用することができる
- クラスの利用方法について理解し，フィールド・メソッドを利用することができる

Processingとは

- イメージ・アニメーション・インタラクションを生み出すソフトウェアを書くために開発されたプログラミング言語
- Javaをベースに開発されたため，オブジェクト指向プログラミングによる開発が可能
- クロスプラットフォーム
  - Windows
  - Linux
  - Mac OS
- https://processing.org/

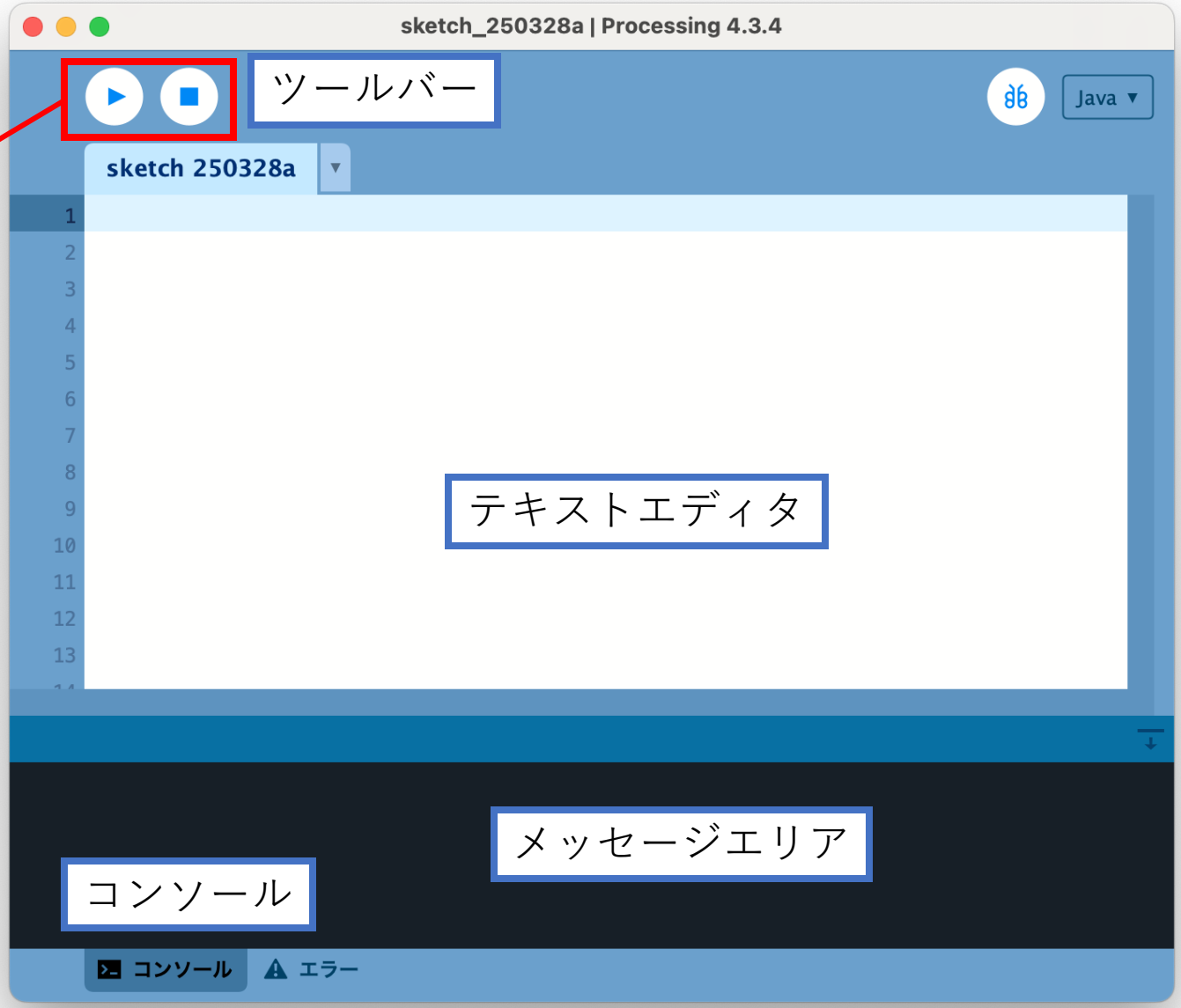


開発環境

- Processing4

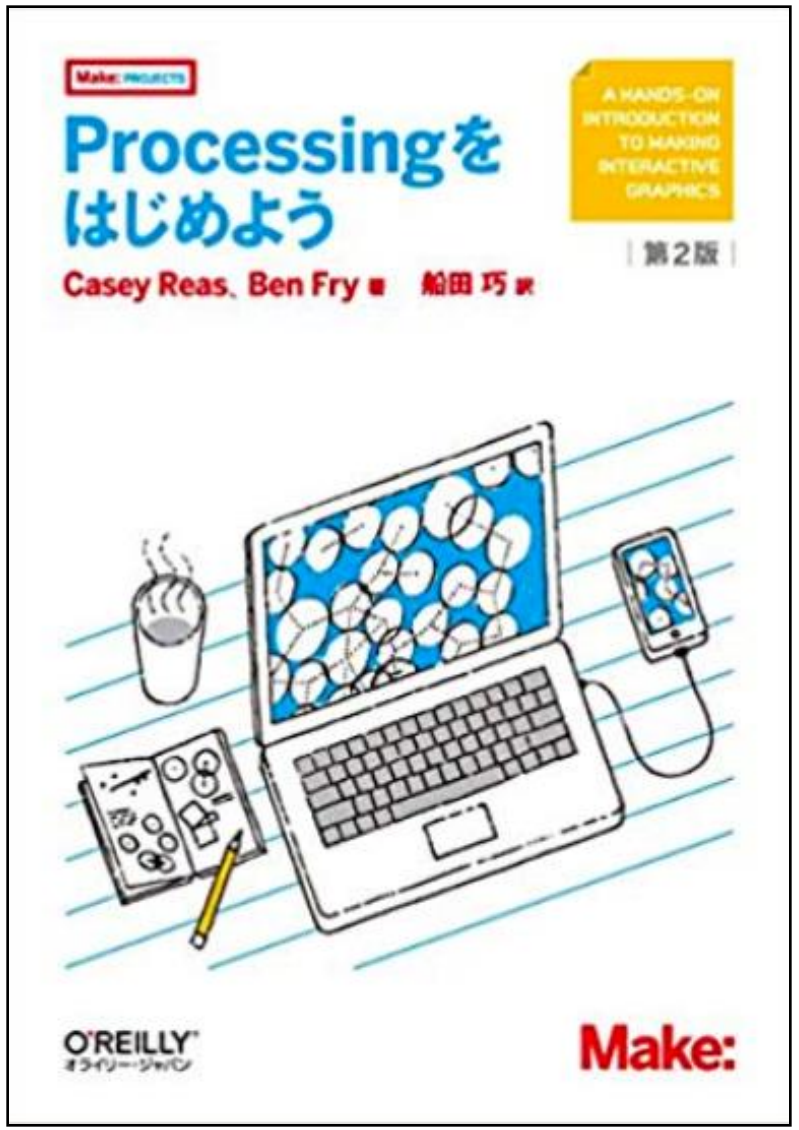
実行ボタン　　停止ボタン

実行ウィンドウ



Processingの参考書

- Casey Reas, Ben Fry, "Processingをはじめよう 第2版," オライリー・ジャパン, 2016.



Processingの基本

- setup()関数
  - 初期化
  - 一度だけ実行される
- draw()関数
  - 繰り返し実行される
  - １秒間に60回実行
  - アニメーションの描画

HCK01_01.pde

```
void setup(){
  size(480, 120);
}

void draw(){
  if (mousePressed) {
    fill(0);
  } else {
    fill(255);
  }
  ellipse(mouseX, mouseY, 80, 80);
}
```

関数

- size()
  - 実行ウィンドウのサイズを決める
  - 第1引数は幅
  - 第2引数は高さ
- fill()
  - 図形の内部に塗られる色を指定
  - 引数に0～255を指定すると
    対応するグレースケールの色となる
  - カラーを指定することもできる
- ellipse()
  - 座標，幅，高さを指定して円を描く

HCK01_01.pde

```
void setup(){
  size(480, 120);
}

void draw(){
  if (mousePressed) {
    fill(0);
  } else {
    fill(255);
  }
  ellipse(mouseX, mouseY, 80, 80);
}
```

システム変数

- mousePressed
  - マウスボタンが押されているとtrue, そうでなければfalse
- mouseX
  - マウスカーソルの水平方向の位置
- mouseY
  - マウスカーソルの垂直方向の位置

HCK01_01.pde

```
void setup(){
  size(480, 120);
}

void draw(){
  if (mousePressed) {
    fill(0);
  } else {
    fill(255);
  }
  ellipse(mouseX, mouseY, 80, 80);
}
```

本授業で使用する主な関数・システム変数

- background(grey)
  - 背景色を設定する
  - 引数に0～255を指定すると対応するグレースケールの色となる
  - カラーを指定することもできる

- stroke(color)
  - 線の色を指定する

- line(x1, y1, x2, y2)
  - 2点間の座標を指定して1本の直線を描く

本授業で使用する主な関数・システム変数

- map(a, b, c, d, e)
  - aを範囲b-cから別の範囲d-eへ変換する

- colorMode(mode, range1, range2, range3)
  - カラーモードを指定する
  - mode: RGB（赤・緑・青）またはHSB（色相・彩度・明度）
  - range1: 赤または色相の範囲（intまたはfloat）
  - range2: 緑または彩度の範囲（intまたはfloat）
  - range3: 青または明度の範囲（intまたはfloat）

- color(range1, range2, range3)
  - 色を作成する
  - パラメータの値はcolorMode()で設定した値により判断される

本授業で使用する主な関数・システム変数

- loadPixels()
  - 実行ウィンドウ画面全体の色情報を配列pixelsとして読み込む
- updatePixels()
  - 実行ウィンドウを配列pixelsの値でアップデートする

本授業で使用する主な関数・システム変数

- PI
  - 円周率π
- sin(a)
  - aの正弦を求める
- pow(a,b)
  - aのb乗を求める
- abs(a)
  - aの絶対値を求める

本授業で使用する主な関数・システム変数

- keyPressed()
  - キーが押されるたびに呼び出される関数

- key
  - 最後に押されたキーを表すシステム変数

- 各関数およびシステム変数の使い方は各自で調べること

動的メモリ確保

- データ型 [] 変数名;
  - 配列変数の宣言
  - int [] array;

- 変数名 = new データ型 [要素数];
  - 動的メモリ確保
  - 要素数には変数を利用することができる
  - array = new int [10];

クラスの利用

- クラス名 変数名;
  - インスタンス格納用変数の宣言
  - Minim minim;
    クラス　　インスタンス格納用変数
- 変数名 = new クラス名(引数);
  - インスタンスの生成と変数への関連付け
  - 引数はクラスごとに異なる
  - minim = new Minim(this);

インスタンス格納用変数

newキーワードを使用してクラスのインスタンスを生成

インスタンスから使えるもの

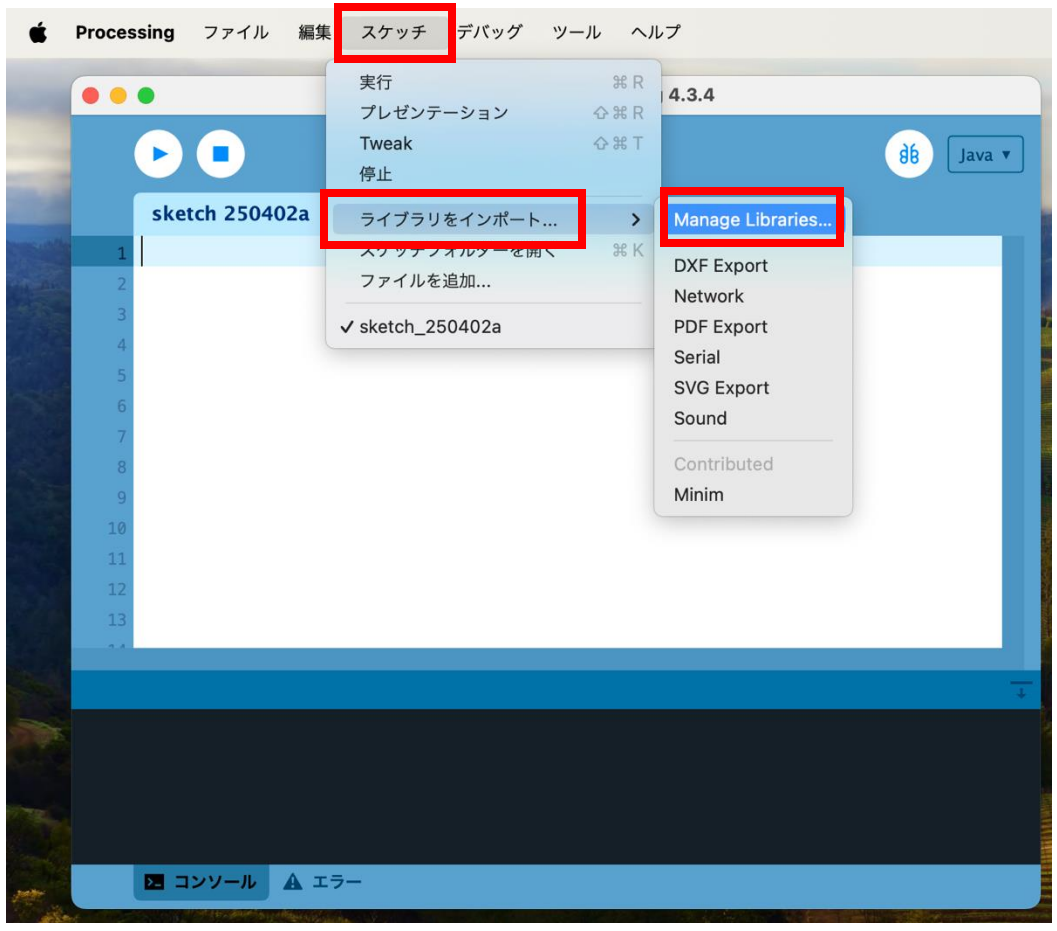
- フィールド（Field）
  - クラス内に定義されている変数
  - インスタンスの変数名にドットをつけて使用する
  - minim.MONO
  - minim.STEREO

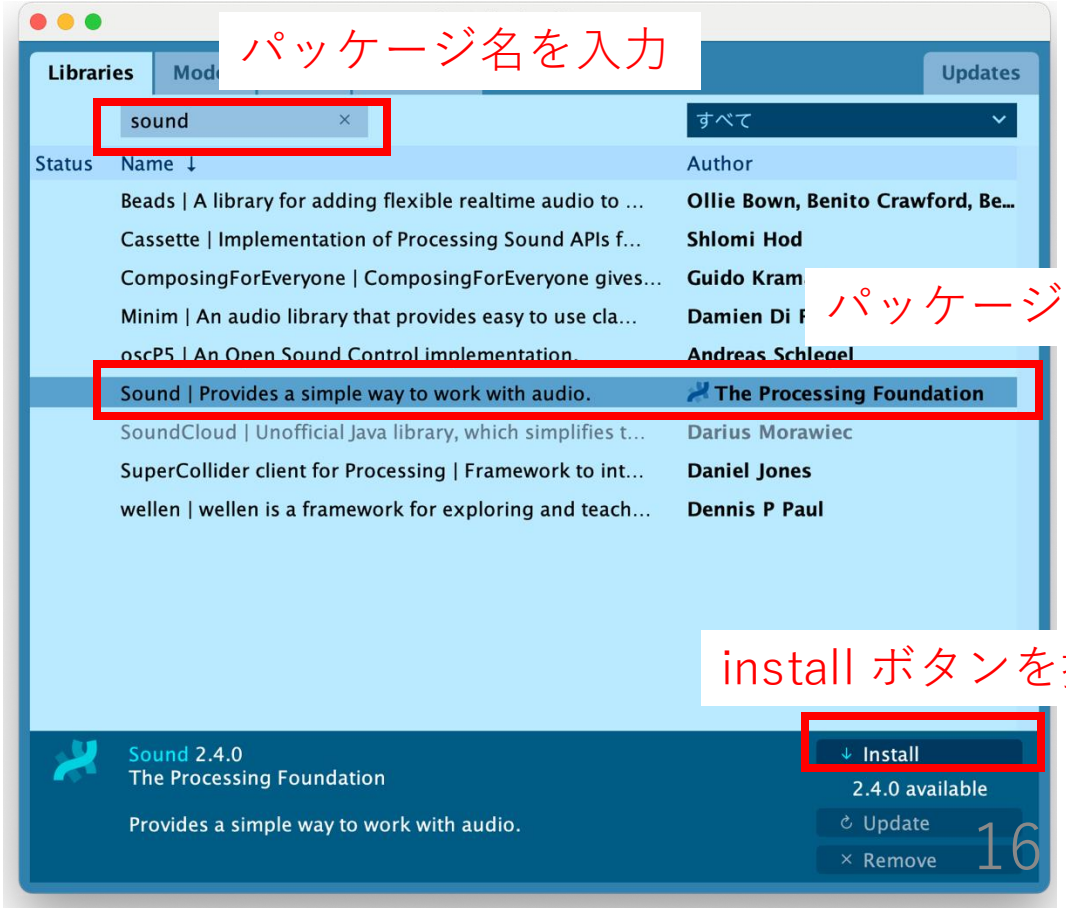
- メソッド（Method）
  - クラス内に定義されている関数
  - インスタンスの変数名にドットをつけて使用する
  - minim.createSample(引数);

- 各クラスにどのようなフィールド・メソッドが用意されているかは各自で調べること

パッケージのインストール

- メニューから「スケッチ」ー「ライブラリをインポート」ー「ライブラリを追加」を選択





まとめ

- 開発環境であるProcessingの知識を習得する
  - 簡単な操作方法
  - 動的なプログラムを書くための基本的な関数の機能
  - 動的メモリ確保
  - クラスの利用方法

- プログラムを実際に動作させて，本資料の内容を確認すること

課題提出

- 動作確認のために作成したプログラムを提出しなさい。
- ファイル名：HCK01_01.pde

- 提出期限：4月9日9時