

プログラミング演習I

第2回 Processingの基礎(1)

2016/04/18



本日の予定

- 「Processingをはじめよう」
 1. ようこそProcessingへ
 2. コードを書いてみよう
 3. 描く



「Processingをはじめよう」の内容

「はじめに」より (ixページ)

- 1章「ようこそProcessingへ」 Processingの基礎について学びます。
- 2章「コードを書いてみよう」 最初のProcessingプログラムを作ります。
- 3章「描く」 基本的な図形を描きます。
- 4章「変数」 データを記憶し、変更し、再利用します。
- 5章「反応」 マウスとキーボードに反応するプログラムを作ります。
- 6章「メディア」 画像、フォント、ベクタ画像などを読み込んで表示します。
- 7章「動き」 図形を意図したとおりに動かします。
- 8章「関数」 新たなコードモジュールを作ります。
- 9章「オブジェクト」 変数と関数が結合したコードモジュールを作ります。
- 10章「配列」 変数のリストを扱いやすくします。
- 11章「拡張」 3次元画像、画像の書き出し、Arduinoボードからの読み込み。
- Processingクイックリファレンス



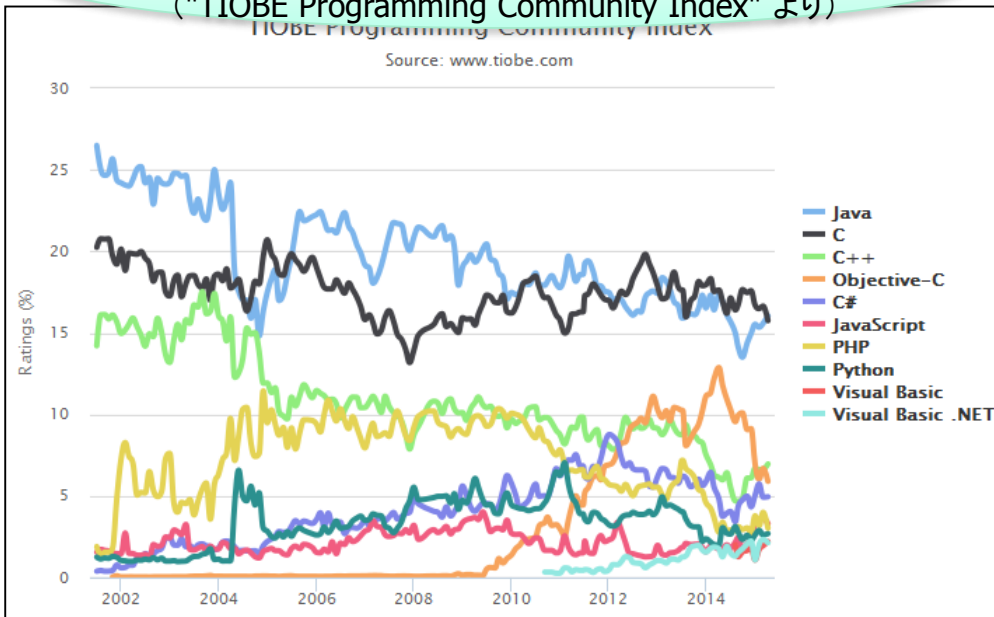
1. ようこそProcessingへ

Processingとは？

- プログラミング言語のひとつ
- プログラムを作るための「開発環境」

人気のプログラミング言語
("TIOBE Programming Community Index" より)

TIOBE Programming Community Index
Source: www.tiobe.com



「開発環境」

The screenshot shows the Processing IDE interface. The code editor contains the following code:

```
sketch_apr13a $  
// 二つの円を描くプログラム  
  
size(480, 120);  
  
noSmooth(); // スムージングを無効に  
ellipse(140, 60, 90, 90);  
  
smooth(); // スムージングを有効に  
ellipse(240, 60, 90, 90);
```

A yellow callout bubble points to the code with the text "プログラムを書いて". Below the code editor, a preview window shows two white circles on a gray background. A second yellow callout bubble points to the preview window with the text "すぐに実行確認できる".



最初のプログラミングにProcessingを使う理由

- わかりやすい (pp.002-003)
 - プログラムの実行結果が絵として表示される(プログラムで絵を描くイメージ)
- 手軽で応用範囲が広い (p.004)
 - 画像や動画、音源などを使うための道具(ライブラリ)が揃っている
- 素性が良い (p.006)
 - Java + 拡張 (Javaでも書ける)
- オープン (p.007)
 - 無料！！ (<http://processing.org/>)






Processingの残念なところ

- 日本語との親和性が低い
- 参考書が少ない



2. コードを書いてみよう

最初のプログラム (p.011)

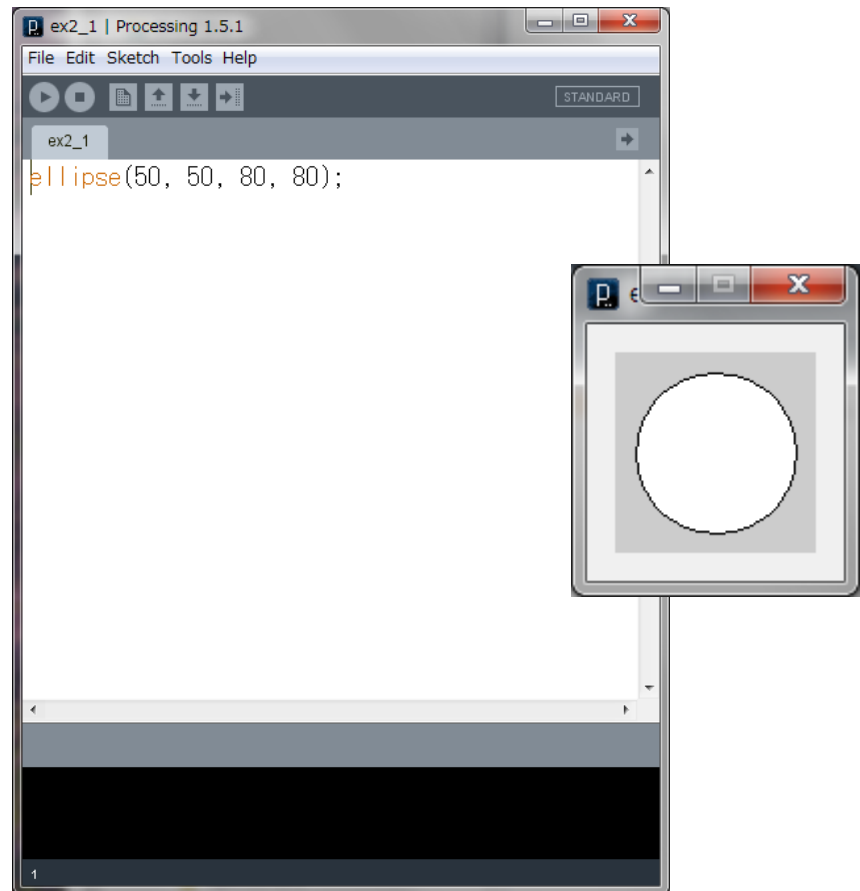
- Processing開発環境 (PDE) を起動
- Example2-1のコードを入力
- 実行 
- 停止 
- プログラムの保存 
 - Zドライブに"Prog1"というフォルダを作る
 - "ex2_1"という名前で保存

注) 名前に使える文字は英数字、アンダースコアのみ

Processing開発環境 (PDE)

(以下を書き込む)

- 実行ボタン
- 停止ボタン
- Saveボタン
- Openボタン
- テキストエディタ
- メッセージエリア
- コンソール
- メニュー
- 実行ウィンドウ





次のプログラム (p.012)

- PDEを起動
- Example2-2のコードを入力
- 実行
 - 実行ウィンドウ上でマウスを動かす
 - マウスをクリックしたまま動かす
- 停止
- プログラムの保存
 - "ex2_2"という名前で保存



コードの書き方

- コードは**英数字、記号のみ**
- 文の最後は必ず**";"**(セミコロン)
- **きれいに書く**
 - 1行に1文だけ書く
 - 区切りに**空行**を入れる
 - 意味の塊(ブロック)ごとに**インデント(字下げ)**を入れる
 - 必要に応じて**コメント**を入れる

コードをきれいに書く

```
void setup() {  
→ size(480, 120);  
}  
  
void draw() {  
→ if (mousePressed) {  
→→ fill(0);  
→ } else {  
→→ fill(255);  
→ }  
→ ellipse(mouseX, mouseY, 80, 80);  
}
```

意味の区切れ目に
空行

ブロック内はすべて
インデント

ブロック内の
ブロックで
さらにインデント

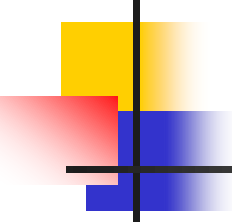


コードをきれいに書く

■ コメントを書く(p.035)

```
// 最初に実行されるモジュール
void setup() {
    size(480, 120);    // ウィンドウサイズ
}

// 1/60秒に一回実行されるモジュール
void draw() {
    if (mousePressed) {
        fill(0);      // マウスが押されていれば黒
    } else {
        fill(255);    // それ以外は白
    }
    ellipse(mouseX, mouseY, 80, 80);    // 円の描画
}
```



開発サポート機能

- エラー表示
- 青文字、赤文字
- オートインデント、オートフォーマット
- アプレット化 (p.014 「シェアしよう」)



サンプルとリファレンス (p.016)

- サンプルプログラム
 - "File"メニューから"Example"
- リファレンスマニュアル
 - "Help"メニューから"Reference"
 - テキストの付録
 - 坂井先生のページ
<http://marble.aomori-u.ac.jp/Processing/>
「Processingに関する情報」



宿題

- 自宅のPCでProcessingを動かしてみよう！