

Design Programming

Spring 2018 - 1

2018.4.11 Keio University, SFC

アンケート

- <https://goo.gl/hHYzP2>

教員紹介

- 青木 聖也

情報科学芸術大学院大学 (IAMAS) 修了.

アーティスト. Scott Allenとして活動中. 投影装置・映像メディアに関する作品を制作している.

慶應SFC政策・メディア研究科後期博士課程社会人コース 脇田研究室所属

<http://scottallen.ws>

- 中嶋 亮介

慶應SFC政策・メディア研究科修士課程 脇田研究室所属

コーディングカルチャーに根ざした表現の研究を行なっている

<http://rystylee.com>

主題と目標／授業の手法

- この授業では、プログラミングを通してデザインについて考えます。デザインといっても、一般的に連想する図や形を美しく配置するグラフィックデザインにとどまらず、図案や形態に加えてサウンドやビデオ、インタラクションを時間軸の中でプログラミングによってデザインしていきます。

授業では、オープンソースのプログラミング言語/開発環境であるProcessingを使用します。Processingを用いたプログラミングの基本から始めて、最終的にはProcessingを用いた3D表現や画像処理、データ解析など、高度なプログラミングによる表現が可能となることを目標とします。

授業予定

- 4月11, 18, 25日
- 5月9, 16, 23, 30日
- 6月6, 13, 20, 27日
- 7月4, 11, 18日

提出課題・試験・成績評価の方法

- 出席
- 履修態度
- 通常課題の内容
- 最終課題の内容

を総合的に評価する。

デザインとプログラミング

仕様

体験型

不特定多数に

リアルタイム

大量に

規則的or不規則に

プログラミング



アウトプット

体験

サービス

ビジュアル

グラフィック

サウンド

Design by Numbersについて

- 1999年, MIT Media Labで教鞭をとっていたデザイナーであり教育者のJohn Maedaによって開発されたデザイナーやアーティストのためのプログラミング環境. 100px四方の小さなウィンドウに簡単な命令でリアルタイムにグラフィックを表示することができる.

<https://vimeo.com/72611093>

Processingについて

- 2001年にMIT Media LabにてJohn Maedaの影響を受けた二人の学生Ben Fryらによって開発されたプログラミング環境.

Design by Numbersのスピリットを受け継ぎ、デザイナーやアーティストがプログラミングを学習し、表現手段として用いることを目的として日々進歩している.

JavaをベースとしてOSを問わず動作するクロスプラットフォームな開発環境であり、ウェブブラウザで動作するjavascriptライブラリ「p5js」も存在する.

<http://processing.org/>

<http://p5js.org/>

Processingを使った作品



作品例

- Creative Applications:

<http://www.creativeapplications.net/category/processing/>

- open processing:

<http://www.openprocessing.org/>

インストール方法

- Processingのウェブサイトから「Download」ページに移動し、自分の環境にあったProcessingをダウンロードする。バージョンは数ヶ月に一度更新される。

全て無料で利用可能だが、日々進歩しているため、下位のバージョンで動いたプログラムがバージョンアップとともに動作しなくなることもあるので注意。

<https://processing.org/download/>

チュートリアル

- Processingには基本的な使い方から、高度な応用表現に至るまで、詳しい説明を含んだチュートリアルが存在する。
公式の「tutorial」にはトピックごとにチュートリアルが整理されている。

<https://processing.org/tutorials/>

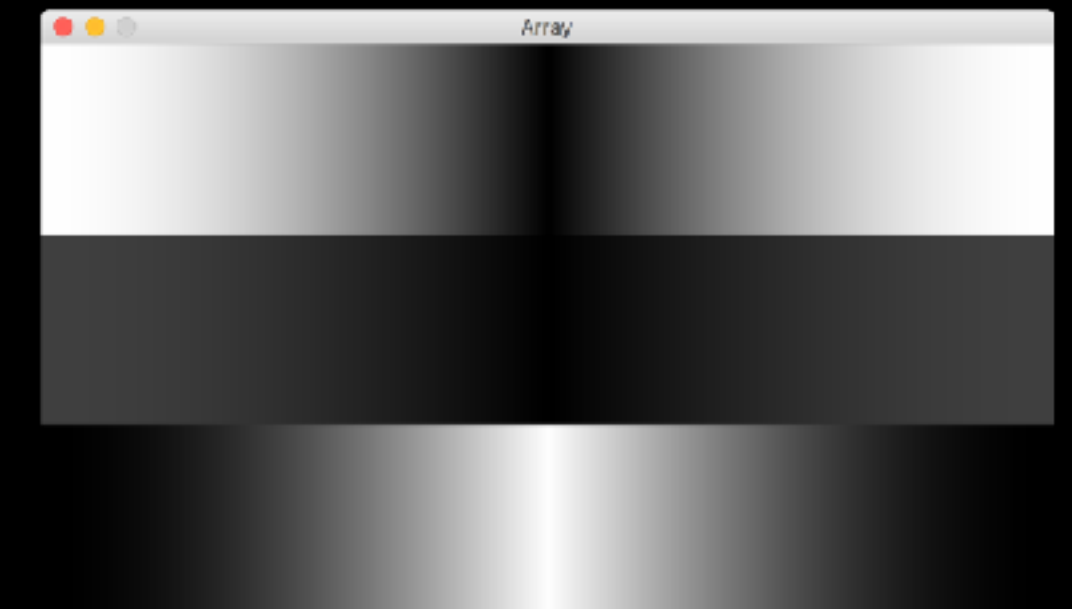
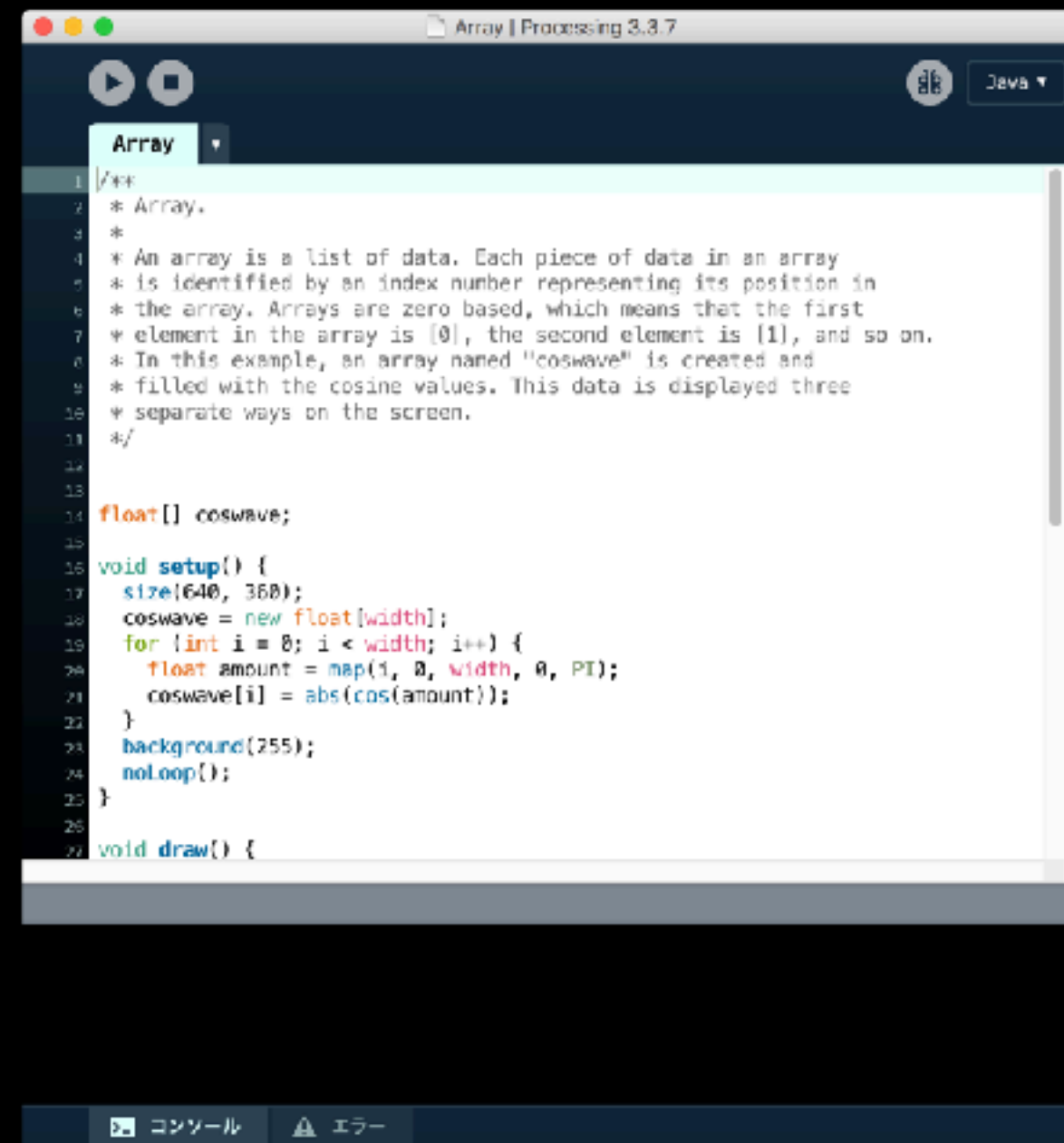
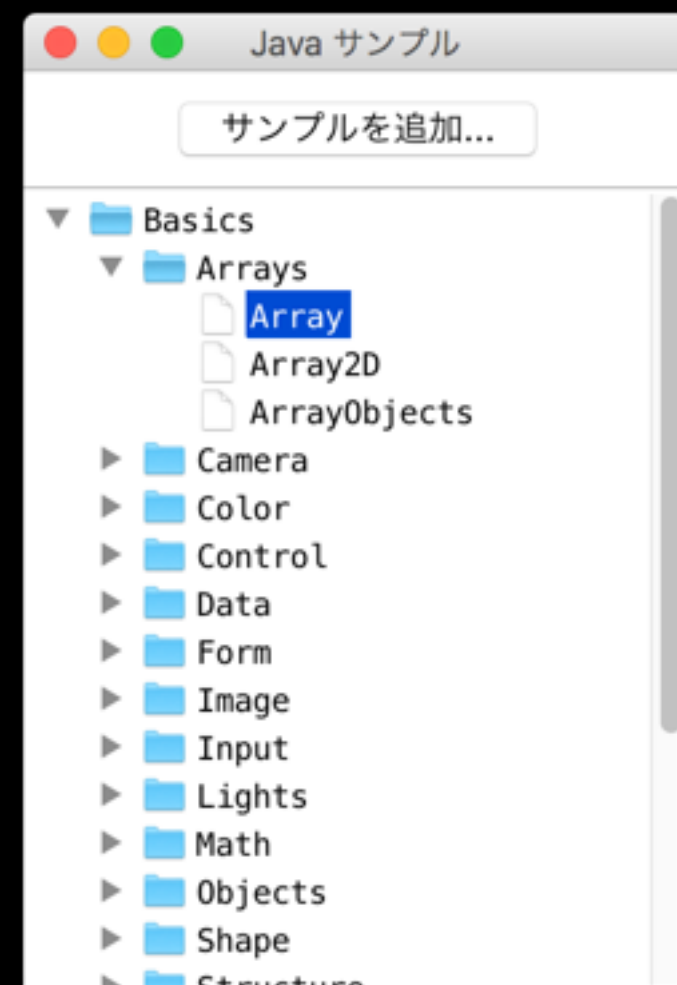
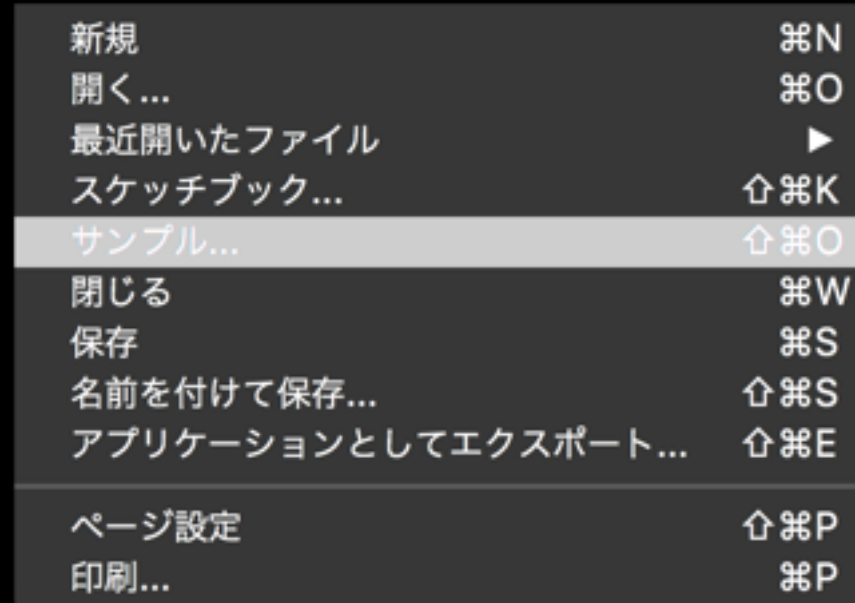
Processing Library

- Processingには標準搭載されていない機能もライブラリをインストールすることで機能を拡張することが可能.
- <https://www.youtube.com/watch?v=DIXDgkE98Ik>



サンプルコード

- ファイル>サンプル



参考書籍

- 「Processingをはじめよう」 オライリー・ジャパン 2011年
- 「Processing: ビジュアルデザイナーとアーティストのためのプログラミング入門」 BNN新社 2015年
- 「ジェネラティブ・アート—Processingによる実践ガイド」 BNN新社 2014年
- 「Generative Design —Processingで切り拓く、デザインの新たな地平」 BNN新社2016年
- 「Nature of Code- Processingではじめる自然現象のシミュレーション」 株式会社ボーンデジタル 2014年
- 「ビジュアライジング・データ—Processingによる情報視覚化手法」 オライリー・ジャパン 2008年
- 「データ可視化プログラミング入門」 秀和システム 2013年

Open Processing

- Processingで作った作品を共有できるサービス。「デザインとプログラミング」の課題提出などは基本的にOpen Processing内で行なう。

<https://www.openprocessing.org/>

Open Processing

- Open Processingのアカウントを作成する.
 1. 「Join」 ボタン
 2. 「NAME」 に好きな名前（実名でなくてOK） を入力
 3. 「EMAIL」 「PASSWORD」 を入力
 4. 「I' m not a robot」 にチェック
 5. 「Join」 ボタン

この授業用ページに参加する

- 「Design Programming」ページに参加する.
 1. <https://www.openprocessing.org/class/57853>
 2. 「Join」ボタン
 3. コード「2C9EAD」を入力
 4. 「Join」ボタン

アカウントと氏名の関連付け

- <https://goo.gl/m3G4gt>

座標について

- Processingでは全て「ピクセル」で考えます

円を描く

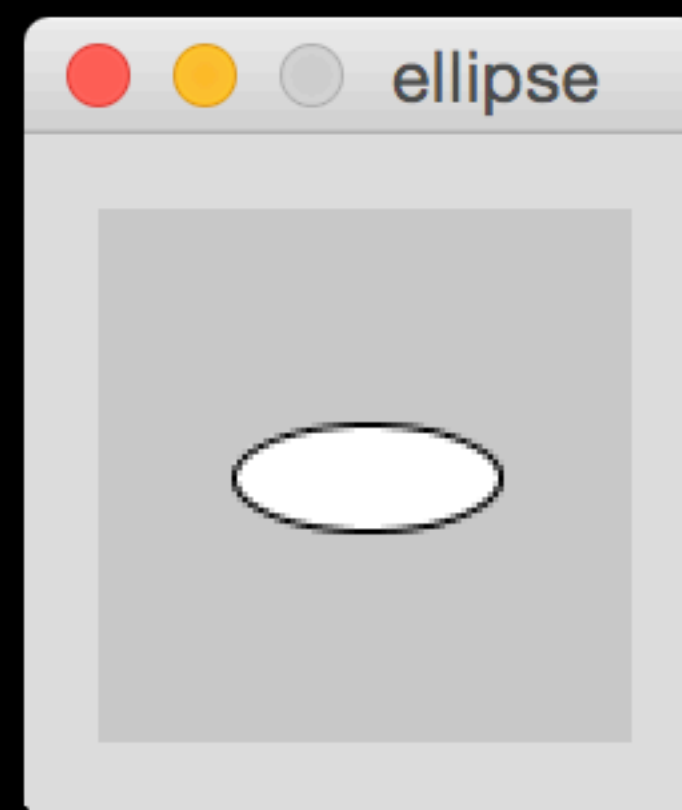
01/ellipse/

```
ellipse(50, 50, 50, 20);
```

楕円 (ellipse) を描く

- 左から50px
- 上から50px
- 幅が50px
- 高さが20px

結果



線を描く

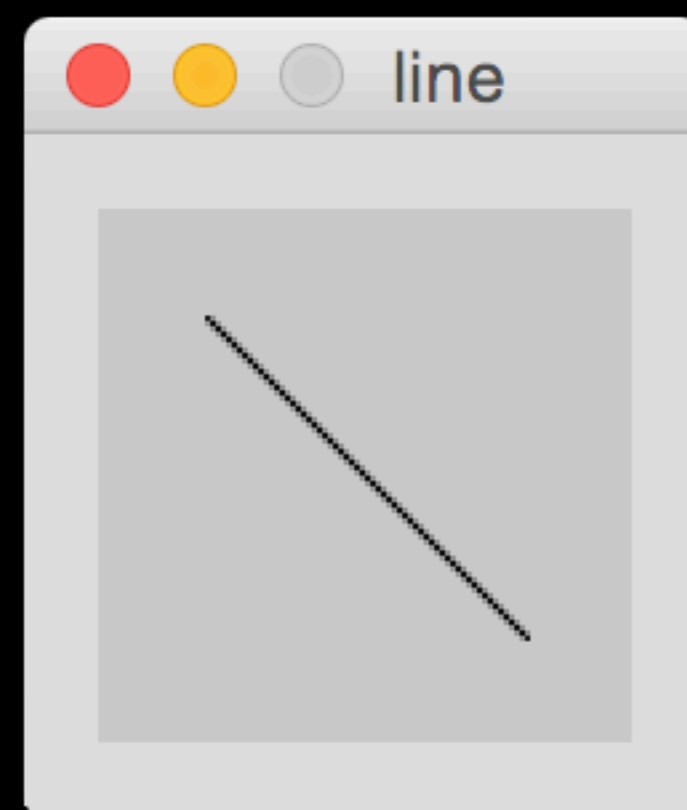
01/line/

```
line(20, 20, 80, 80);
```

線 (line) を描く

- 点(20, 20) から
- 点(80, 80) へ

結果



三角形を描く

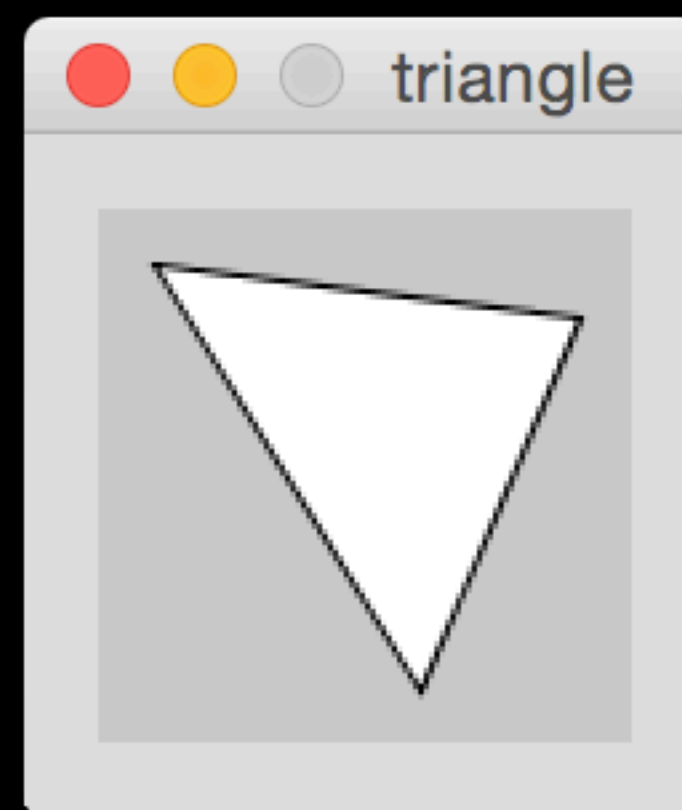
01/triangle/

```
triangle(10, 10, 90, 20, 60, 90);
```

三角形 (triangle) を描く.

- 点(01, 10) と
- 点(90, 20) と
- 点(60, 90) の

結果



四角形を描く

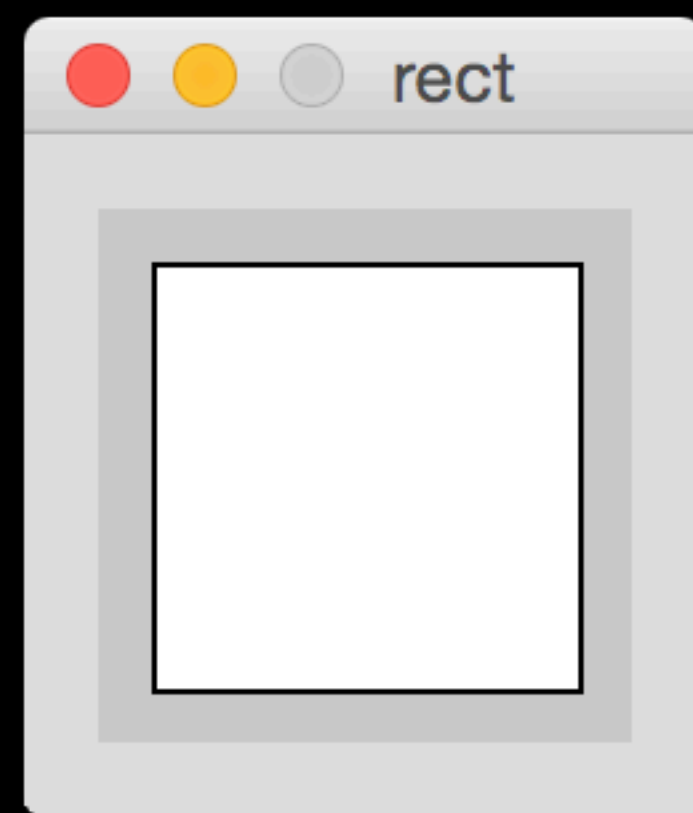
01/rect/

```
rect(10, 10, 80, 80);
```

四角形 (rect) を描く

- 点(10, 10) から
- 横80px で
- 縦80px の

結果



画面サイズを変更する

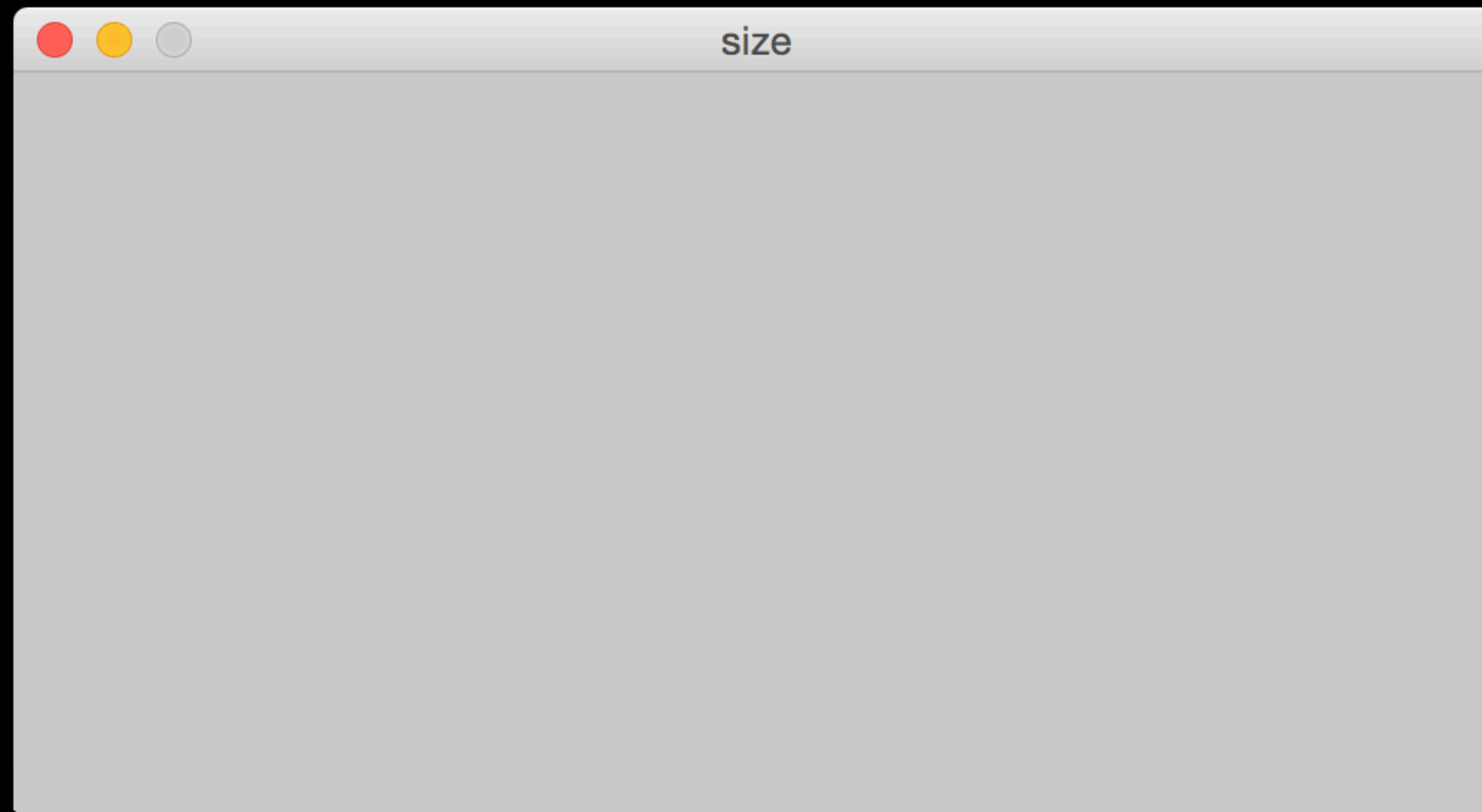
01/size/

```
size(500, 250);
```

画面サイズを変更する

- 横500px
- 縦250px に

結果



線の太さを変更する

01/size/

```
strokeWeight(5);  
ellipse(50, 50, 30, 30);
```

線の太さ (strokeWeight) を変更する

- 線幅5px で

結果



課題

- 自分が面白いと思う作品, 興味を持てる作品をOpen Processingで見つけ「hearts」すること. (最低30個)

〆切: 4月17日24時まで

次回

- Processingの基本
- Open Processingを使ったスケッチの共有