
プログラミング演習 (0)

イントロダクション

中村、高橋
小林、橋本

まずは

- 教員とTAの紹介

次に

- 近くのひとと自己紹介して下さい
 - 感染リスクを考慮して、マスクをして、声は抑え気味でお願いします
 - オンラインの人はごめんなさい

何故 Java や C じゃないの？

- プログラミング嫌いを量産したくないから
 - 最初にJavaやCを学んだ学生さんの多くが挫折し、プログラミング自体を嫌いになってしまう
 - 苦手意識を払拭できず、コンピュータ自体も嫌いに
 - 一度嫌いになったら、そんじょそこらの教え方ではプログラミングできるようになりません...
- 例えば、プログラミングでウィンドウを出すこと自体が大変で、心が折れてしまうものです

JavaやC言語でウィンドウ

Java

```
import java.awt.*;

public class Window01 {
    public static void main(String args[]){
        WindowTest windowText=new WindowTest();
    }
}

class WindowTest{
    WindowTest(){
        Frame frame;
        frame=new Frame("Window Test");
        frame.setSize(240, 240);
        frame.setVisible(true);

        Label label;
        label=new Label("Hellow Window");
        frame.add(label);
    }
}
```

C言語

```
#include <windows.h>
#define WNDCLASSNAME TEXT("SampleClass")
LRESULT CALLBACK WindowProc(HWND hWnd, UINT msg,
{
    switch (msg) {
        case WM_DESTROY:
            PostQuitMessage(0);
            return 0;
    }
    return DefWindowProc(hWnd, msg, wp, lp);
}

int WINAPI WinMain(HINSTANCE hInst, HINSTANCE hPrevInst, PSTR lpCmdLine, int
nCmdShow)
{
    HWND hWnd;
    WNDCLASS wc;
    MSG msg;
    wc.style          = CS_HREDRAW | CS_VREDRAW;
    wc.lpfWndProc     = WindowProc;
    wc.cbClsExtra     = 0;
    wc.cbWndExtra     = 0;
    wc.hInstance      = hInst;
    wc.hIcon           = LoadIcon(NULL, IDI_APPLICATION);
    wc.hCursor         = LoadCursor(NULL, IDC_ARROW);
    wc.hbrBackground  = (HBRUSH)GetStockObject(WHITE_BRUSH);
    wc.lpszMenuName    = NULL;
    wc.lpszClassName  = WNDCLASSNAME;

    if (!RegisterClass(&wc)) return 0;
    hWnd = CreateWindow(
        WNDCLASSNAME,
        TEXT("SampleWindow"),
        WS_OVERLAPPEDWINDOW | WS_VISIBLE,
        CW_USEDEFAULT, CW_USEDEFAULT, CW_USEDEFAULT, CW_USEDEFAULT,
```

正直心が折れます

もちろん

- Java や C言語の良いところは沢山
 - 高速に動作する
 - 色々な環境での開発に利用される
 - Windows, Apple, Linux, Web, SmartPhone, ...
 - ライブラリ（サポートしてくれるもの）が豊富
 - 多くの人が開発に利用している

- 2年次にJavaは扱う予定

何故 Processing か

- 視覚的なプログラム作成が容易
- プログラムの基礎を学ぶのに適している
- 最先端の研究でも実を言うと使われている
 - FMS学科の研究でもかなり使われています
- Processing は Java の上で動作しているものであり、Java にととても似ている
- Processing を勉強した後、Java を勉強するのはとても簡単！

超重要

- ウェブ上のプログラムや他人のプログラムをコピーして使うことは厳禁！
 - プログラミング演習はプログラムが書けるようになることを目的としています！
 - 写すと時間とお金の無駄です！！
- 他人のプログラムを写しているような行為が発見された場合は、提供した人も、コピーした人も両者をカンニング扱いとします
 - コピーする人は本当に稚拙な書き換えしかしませんので要注意！！
- 「見せない」「写させない」が重要です

何故？

- X年前の約25%の学生がコピーに該当
 - もちろん、悪意のあったものはここまで多くはありませんが、そういうのを普通であるにとらえている人が多いという事実がありました
- 他の講義でもレポートのコピーが問題に

プログラミング

- ある目的を達成するプログラムの書き方は山のように存在している
 - ある積み木セットを利用して、4つの塔がある城を作れという課題が与えられたときに同じものができらるだろうか？
- また、プログラミングは文章を書くことと似ているため、そこには個性が現れる
 - 文章が一致することはとても少ない

コピーとは？

- 他人から完成したプログラム自体を受け取り、それに何らかの手を加えること提出
- 他人から完成したプログラムの写真を受け取って、それを写して手を加えること提出
- 他人に完成したプログラムをディスプレイに表示し続けてもらい、それを写して手を加え提出

思っている以上に . . .

- みなさんが思っている以上に、コピーさせてくれという人は稚拙な書き換えしかやりません
 - 変数名変更、数値の値変更（小説で登場人物の名前を変えたようなもの）
 - コメントの追加・削除・変更
 - その割に、特殊な変数や記述を使い続ける
- そもそももっとお馬鹿なことも
 - 他人のプログラムをそのまま出している

情報技術を使えば簡単に近さも計算できます

提供してもらわないように！

- みなさんがコピーしてしまうことにより、コピー提供者も迷惑を被ります
- この講義を前提に、
 - エンタテインメントプログラミング演習（1年夏）
 - プログラミング演習2（1年秋）
 - メディアプログラミング実習（2年春）
 - アルゴリズム基礎（2年春）
 - Webプログラミング実習（2年秋）
 - があるため、逃げたらずっと苦しむだけ。勉強にもなりませんのでしんどいなりに立ち向かいましょう！
- これまで英語や数学とかを勉強してきたときどうしています？ どれくらいの時間をかけてきました？

提供しないように！

- 良かれと思ってやりたくなる気持ちはわかります
- 教えるのが面倒でこれみとけって気分になることもあるでしょう
 - でも、同一レポート（課題）の提出があった場合、提供された側が正直に申告しない場合は提供元・提供先の両方が処罰の対象となります
 - そんなしょうもないことで人生を狂わせてはもったいないので、くれぐれも注意してください
- 他の講義でも問題になりつつあります。絶対にレポートはコピーさせないこと！

これまで来た質問

- 教えたら類似するのでは？
 - 多少は類似するでしょう。ただ、積み木でお城を作るという課題で同じ形、同じ色の積み木を同じように積むことはあり得るでしょうか？
 - 見せなければ一致しない
- 教えることを禁止するのか？
 - ぜひとも教えて下さい、ただコピーが発生するような行為は慎んでください
- 他人のプログラムを見てはだめなのか？
 - 教えるときは見ざるを得ませんので、その時は見て下さい。ただ、見せて教えないで下さい。

さて、プログラミングとは

- カップラーメンを作る方法をどう伝える？
 - 「お湯入れて3分待つだけ」
 - で、ちゃんと伝わるかどうか？

さて、プログラミング習得

- 習得する方法には主に、2つの道があります

(1) 作りたい欲にまかせて、自分で調べつつ色々なプログラムを作っていく

- 本当はこちらが上達速度も早くおすすめですよ
- ただ誰もがができる道ではないとはわかっています

(2) ひたすら写経していく

- 作りたい欲がない場合、一番手っ取り早いのが勘所を掴むため写経するということになります。ひたすら書いて覚える！
- そのためにシステムも用意しました `typing.run`

– 他で、競技プログラミングをやる手もあるよ

ということで

- 「作りたい欲」がでないと思う人は、まずは `typing.run` を利用してプログラムを何度も写経してみてください
 - そのうち、なにか作りたいものが見えてくる人も多いものです
 - また、予習資料にあるプログラムや課題、過去の課題などにも挑戦してみましよう！
- タイピング練習と思って、まずはプログラムをどんどん書いてみましょう！

ちなみに

- なぜ `typing.run` をやるの？
 - プログラミングでつまづきやすいポイント
 - 英語アレルギー
 - 命令名のケアレスミス
 - `{ } () , ;`などの記法
 - タイプ速度が遅い
 - こうした問題をタイピングにより解決
 - 英語になれる
 - 命令名の入力ミスはしない
 - 記法になれる
 - タイプ速度が早くなる

進め方

- 講義資料は事前にPDFで配布
 - PDFは必ず予習しておくこと
 - 資料は <http://nkmr.io/lecture/>
 - 本来「小テスト」を講義の最初に実施していたのですが今年度は途中どうなるかわからないので小テストはなし
- 課題を配布
 - 基本課題と発展課題の両方を紙で配布
 - 説明の後、各自課題に取り組む
- 課題を共有フォルダに提出
 - 教員とTAで課題のチェックを実施
 - 基本課題が終わるまでは帰ってはだめ！
 - 発展課題は講義中にはチェックしませんので確実に動作しているかどうかを周りのひとなどと確認しましょう！

講義の流れ

- 13:30-13:50 解説
- 13:50-14:00 課題提示
- 14:00-16:30 課題を解く + 提出する
- 16:30-17:00 課題解説

評価（コロナの状況次第では変更も）

- **各回のタイピング予習の達成度：20点**
 - 各回の課題をすべて課題の目標を達成して満点
 - 予習の点数については次ページで
- **基本課題：40点**
 - 基本課題は必ず終わらせて提出してください
 - 時間内に採点します
- **発展課題：10点**
 - 発展課題は時間内に完成させると加点されます
 - 時間内には採点しません
- **最終課題：30点**
 - 最終試験は実施できないため、課題提出で代替します

タイピング予習

- 各回の予習点数
 - 予習点数 = 各回の点 * (達成課題ポイント合計 / 課題数)
- 1ポイントとなる条件
 - その課題で1度は250CPM以上を達成した
 - その課題で1度は225CPMを達成しており2回タイピングした
 - その課題で1度は200CPMを達成しており4回タイピングした
 - その課題で1度は175CPMを達成しており6回タイピングした
 - その課題で1度は150CPMを達成しており8回タイピングした
 - その課題で10回タイピングした

タイピング予習

- 達成課題ポイントは0 or 1ではなく、実数値とします
- あるCPMを達成している場合に、n回がノルマとしてあげられている場合で、m回実施していた場合、 m/n が達成課題ポイントとして付与されます
 - 現在、第1~3回分については、過去のノルマが生きてるので少し表示がおかしいです（後ほど修正される予定です）。

(1) 円を画面の中央に表示したい！

画面の中央に配置したい！（横幅と縦幅を2で割った値が中央だ！）

430CPM (2回) | 1位/50人中
typo min:3 max:5 avg:4

[edit](#)

(2) マウス位置に合わせて円を2つ描画するには？

計算を利用してマウスの位置に合わせて円を2つ描画しよう

411CPM (2回) | 2位/44人中
typo min:0 max:4 avg:2

[edit](#)

(3) 円を繰り返し動かしたい

300x200のウィンドウで円を画面の左端から右端まで移動し、右端まで来たら左端に登場させたい

390CPM (1回) | 2位/40人中
typo min:4 max:4 avg:4

[edit](#)

(4) 左上から右下に円を動かそう

左上から右下に円を動かし、その値の変化を見てみよう。

366CPM (3回) | 1位/38人中
typo min:5 max:7 avg:5.7

[edit](#)

(5) ローカル変数とグローバル変数

それぞれの変数で円を描画して、グローバル変数とローカル変数の振舞のの違いを知る

402CPM (1回) | 2位/25人中
typo min:3 max:3 avg:3

[edit](#)

(6) 四角形と面積を計算

左上の原点からマウスカーソル位置まで四角形を表示し、その面積を計算して表示する！

383CPM (2回) | 1位/32人中
typo min:5 max:10 avg:7.5

[edit](#)

Typing.runの使い方

- Meiji mailでログイン
 - タイピングできることを試して下さい
 - (1) 最初のプログラムを入力しましょう！
- 自宅から接続する場合は、VPN接続をする必要があります！
 - 読めば多分分かるはず！
 - <https://www.meiji.ac.jp/mind/ssl-vpn/>

今回の目標

- 終わった部分
 - Typing.run の使い方を学ぶ
- これから
 - 20～30分程度でパソコンの使い方（高橋）
 - Processing で入力してみる
 - Google Formで提出する練習をする

パソコンの使い方

- 基本のお話

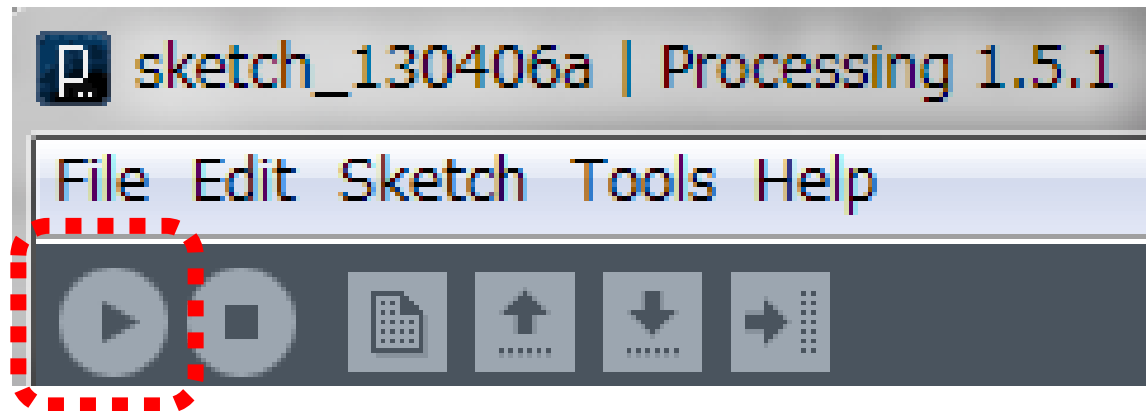
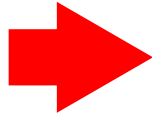
なんかしらんけど

- 自宅からでも VPN 使わずに入れるっぽいです。どうも今はセキュリティ緩め設定？
- Typing.run で第1回について、達成しているのに1ptいってないのは去年の設定が残っているせいですので気にしないで下さい

Processing とは

- Processing プロジェクトは2001年春に開始
- アート、デザイン界向けにJava拡張として
- プログラムすることをスケッチするという
- 多くのライブラリを導入することで各種の処理が可能に
- プログラムの実行は再生ボタンを押すだけ

再生ボタン



想像してみましよう

- 画面にウィンドウを作るのってどうする？
- 国旗のようなものはどうやって作る？
- どうやったら沢山の平行な線を描ける？
- キャラクタはどうやって描画する？
- キャラクタを動かすにはどうしたら良い？

最初に

- 下記のコード（プログラム）を入力して、再生ボタン（実行ボタン）を押してみましょう

```
size(400, 300);
```

400x300のウィンドウを作る

```
background(255, 255, 255);
```

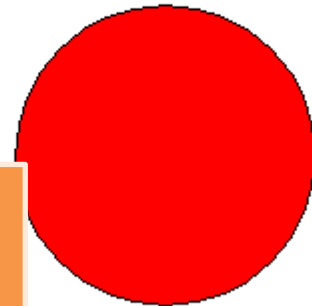
背景色を白色(255,255,255)にする

```
fill(255, 0, 0);
```

塗りつぶしの色を赤色(255,0,0)にする

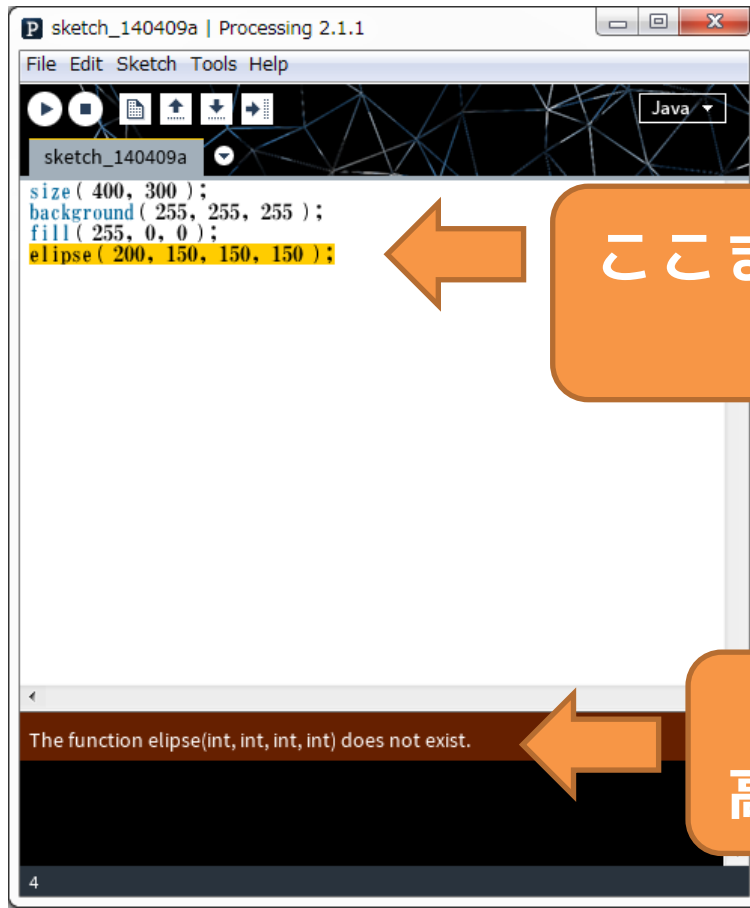
```
ellipse(200, 150, 150, 150);
```

(200,150)を中心として横直径と縦直径150の円を描く
塗りつぶしの色は最後に指定した赤色



動かない??

- 何かプログラムが読み込めないエラーが発生した時、Processingはハイライトしてそこら辺がおかしいということを教えてくれます



ここまたは、ココらへんがおかしい
という意味

エラーメッセージ
高校レベルの英語なので読もう

動かない??

The function `ellipse(int, int, int, int)` does not exist.
(訳) `ellipse(int,int,int,int)`という関数はないよ!



ellipse は ellipse の間違い!

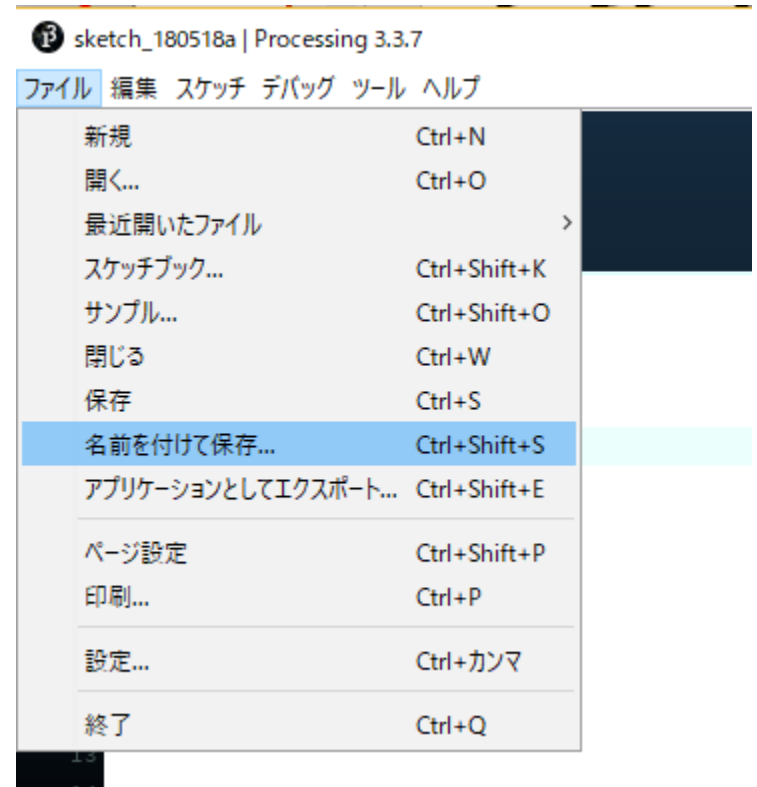
**注意してみると ellipse は青色に
ellipse は黒色になっており違う!**

エラーは最初に注目

- エラーメッセージは沢山表示されます
 - 一部おかしなところがあると、そこから他の部分もおかしいと判断されることがよくあります
 - 表示されるエラーメッセージは、最初にどんなメッセージが表示されているか注目しましょう
- ここにエラーがある！と完璧に推定することはコンピュータには難しい
 - その行または、その前後の行におかしなところがないかをチェックしよう！
 - 例えば、セミコロンが抜けている場合に、次の行でエラーが出ます

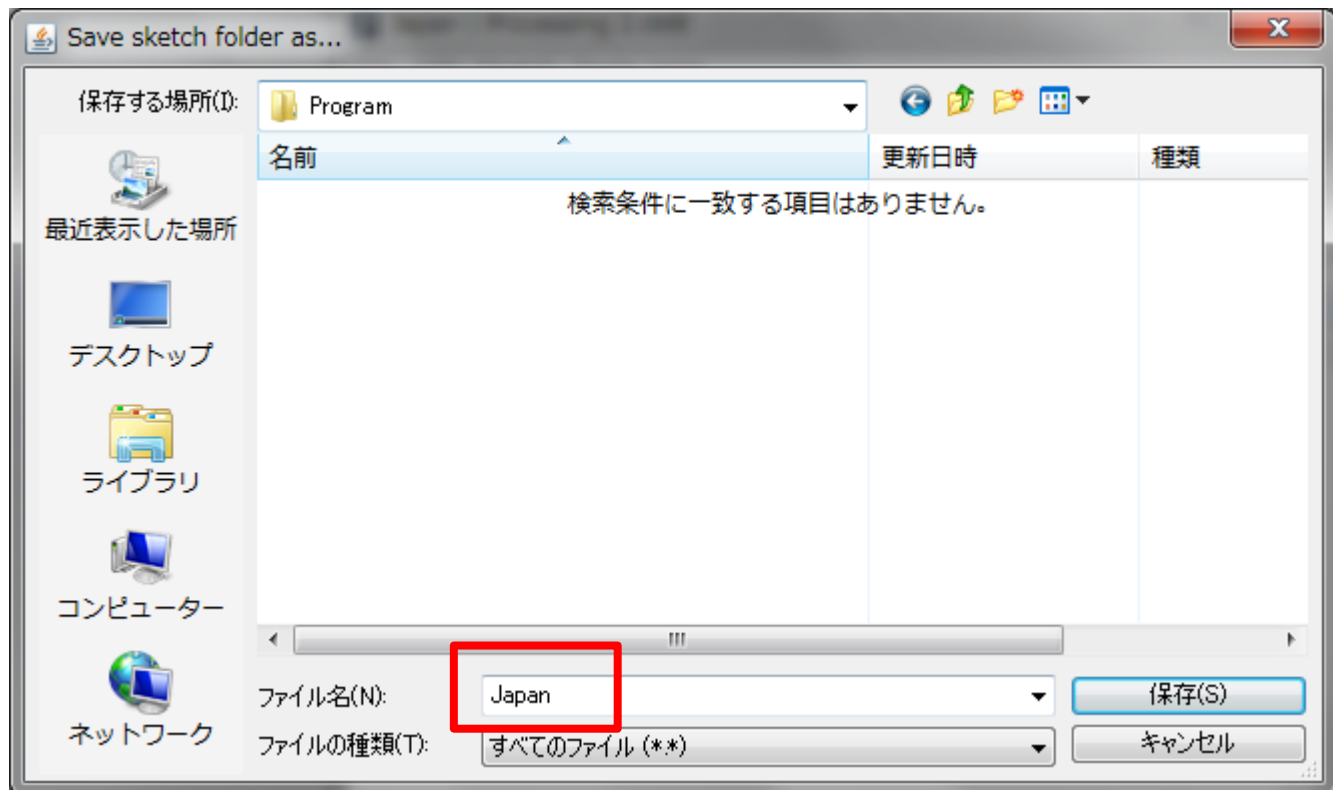
保存しましょう

- ファイル → 名前を付けて保存 で保存
 - 名前をつけて保存
 - どんどん保存しましょう
- ファイル → 新規
 - プログラムを新規作成



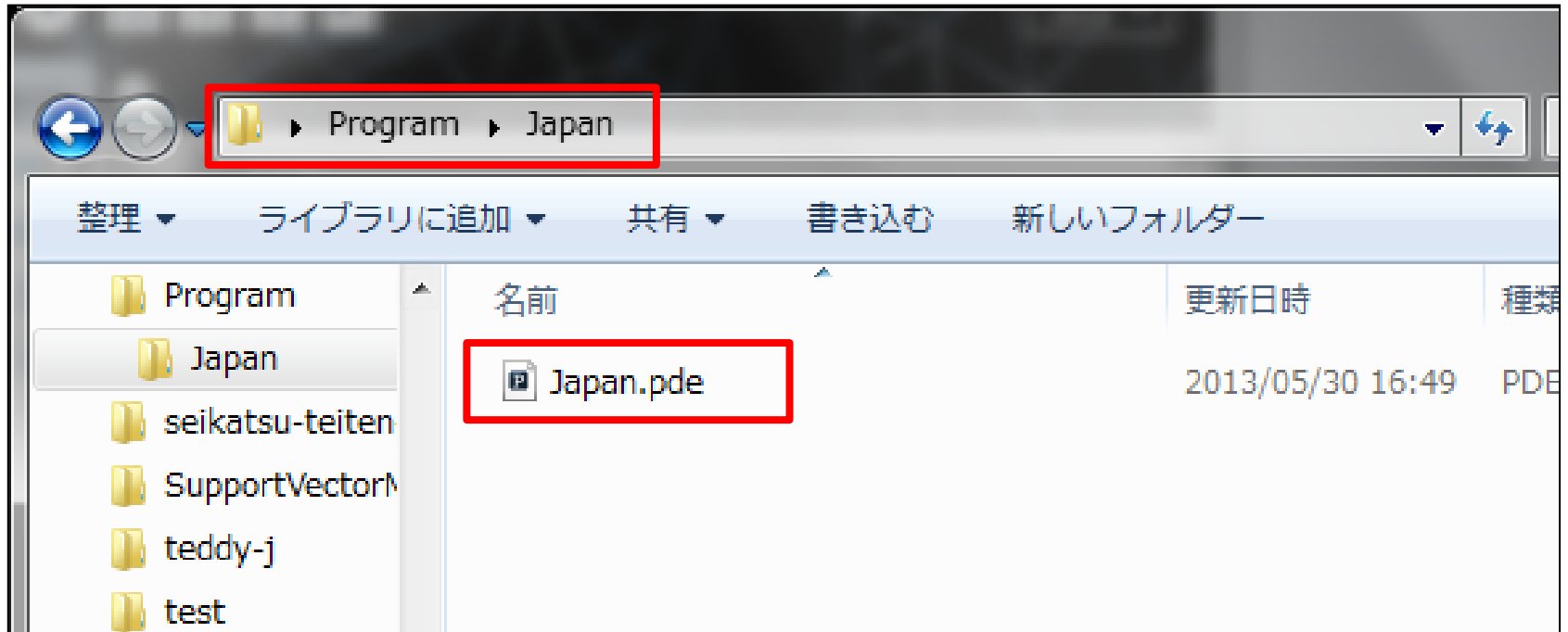
フォルダ構造

- Processingのプログラムは、フォルダとセット
 - (例) スケッチ(プログラム)を、デスクトップにProgramというフォルダを作成し、「Japan」という名前で保存する場合 (WindowsもMacも同じ)



フォルダ構造

- 「Japan」という名前で保存すると。。。
 - 下図のように、名前が「Japan」というフォルダの下に、「Japan.pde」というプログラムが保存される
 - Japanというフォルダは自動生成される

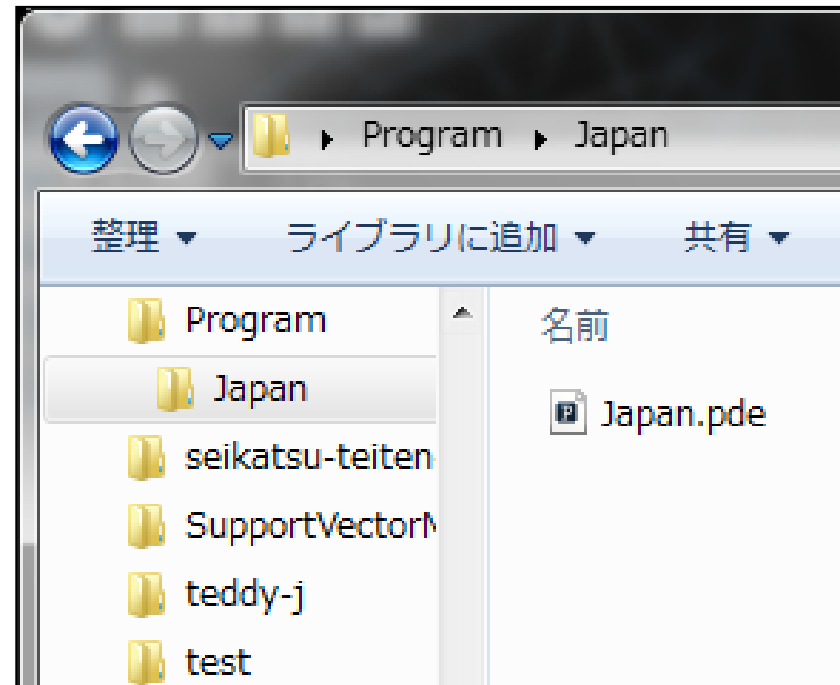
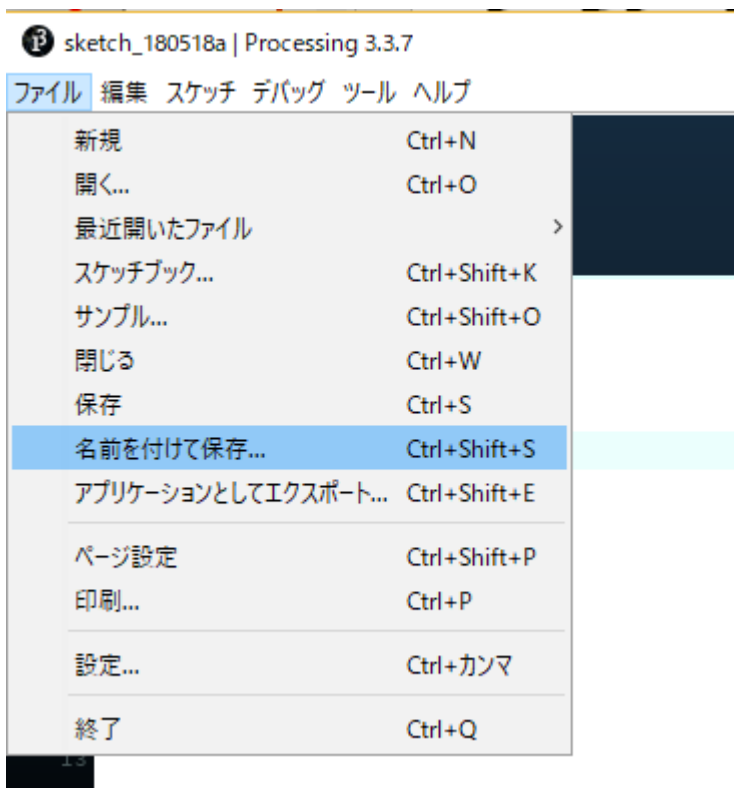


注意点

- Processingの名前は英数字のみ（ただし、1文字目は英文字のみ）
 - Processingでは、日本語の名前でプログラムを保存することはできません
 - Processingでは、プログラム名に空白や記号は使えません

開き方

- メニューの ファイル→ 開く から選んで開く！
- 関連付けを行うとダブルクリックで開けるようになる！



初回の課題

- Google Form からCheckを出す
 - 組・番号・名前
- Gmail のアドレスを提出する
- Processing で日本の国旗が表示できてプログラムを保存する
- Slack にログインして「#講義-プログラミング演習2021」のチャンネルに入り、氏名が「B1-組-番号 名前」、表示名がアルファベット表記になっていることを確認する
 - ブラウザからは `fms-all.slack.com` へ
- `typing.run` にログインして、第1回の4つのプログラムを最低1回入力する（自己申告）

インストール方法

1. Processingを下記のURLよりダウンロード

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

Download Processing. Processing is available for Linux, Mac OS X, and Windows. Select your choice to download the software below.



3.3.7 (13 March 2018)

[Windows](#) 64-bit

[Windows](#) 32-bit

[Linux](#) 64-bit

[Linux](#) 32-bit

[Linux](#) ARMv6hf

[Mac OS X](#)

インストール方法

2. ダウンロードしたファイルを解凍（展開）
3. Processing.exe を起動（Windowsの場合）

Mac は Processing を実行
Linux は % sh processing

