

座標変換、変数の有効範囲、関数



PROCESSING FRIENDS



# 今日の内容

---

座標変換

変数の有効範囲

関数

プログラムが動かない-Σ、(´Д｀;)ノうおおお！

# 座標変換

---

## 座標変換

現在の座標軸の状態を基準に  
描画が行われる

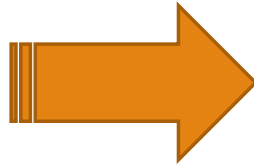
座標軸を移動させる

translate

rotate

scale

座標軸



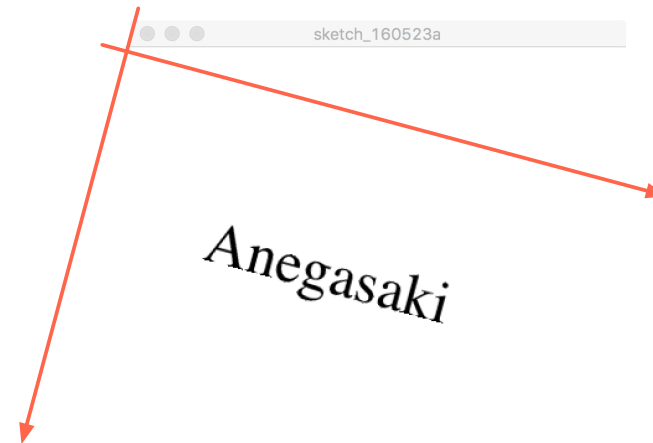
図形や文字列を表示  
する時の基準

```
text("Anegasaki", 100, height/2);
```

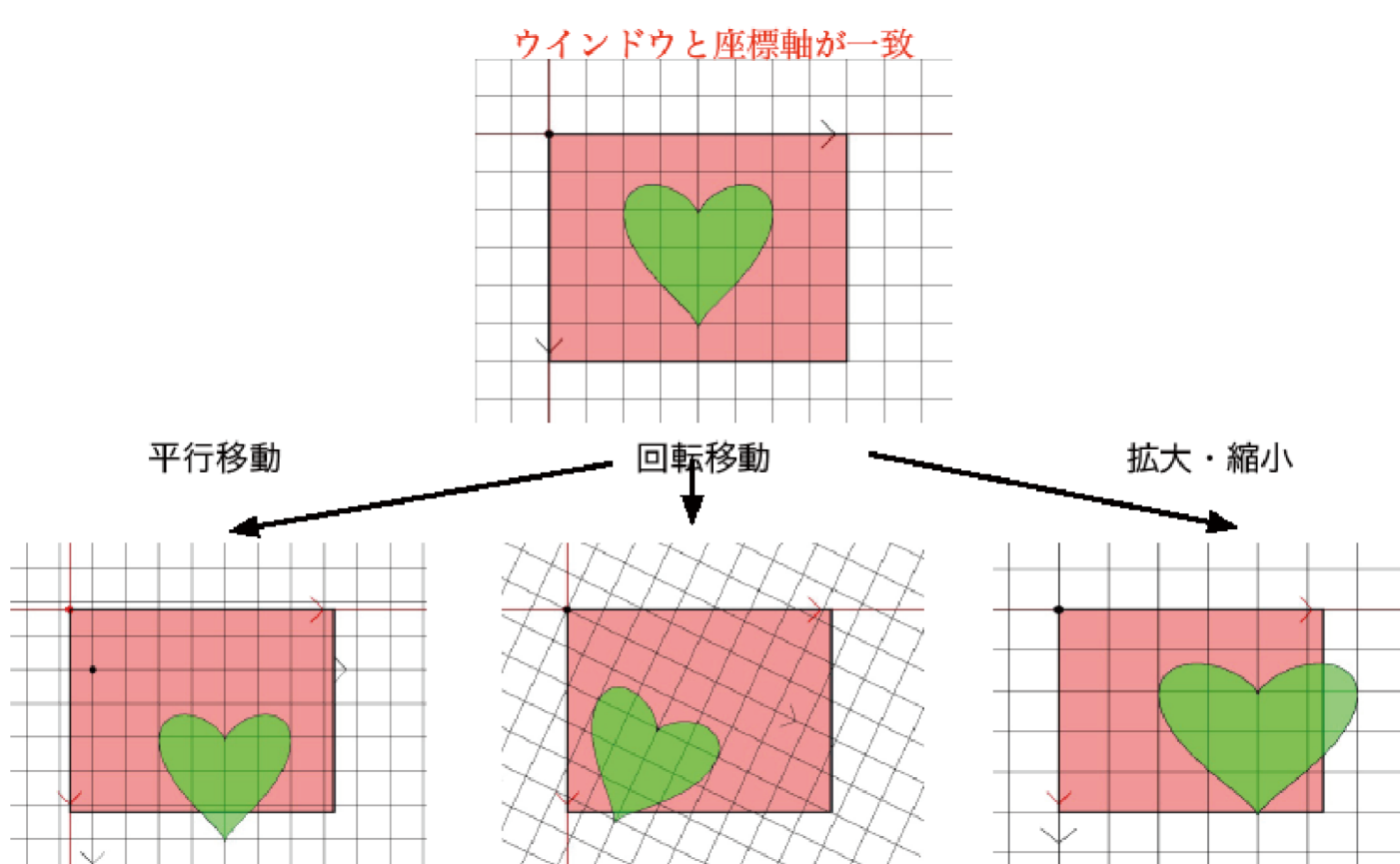
デフォルトの座標軸



座標軸を15度回転



# 座標変換



# 座標変換

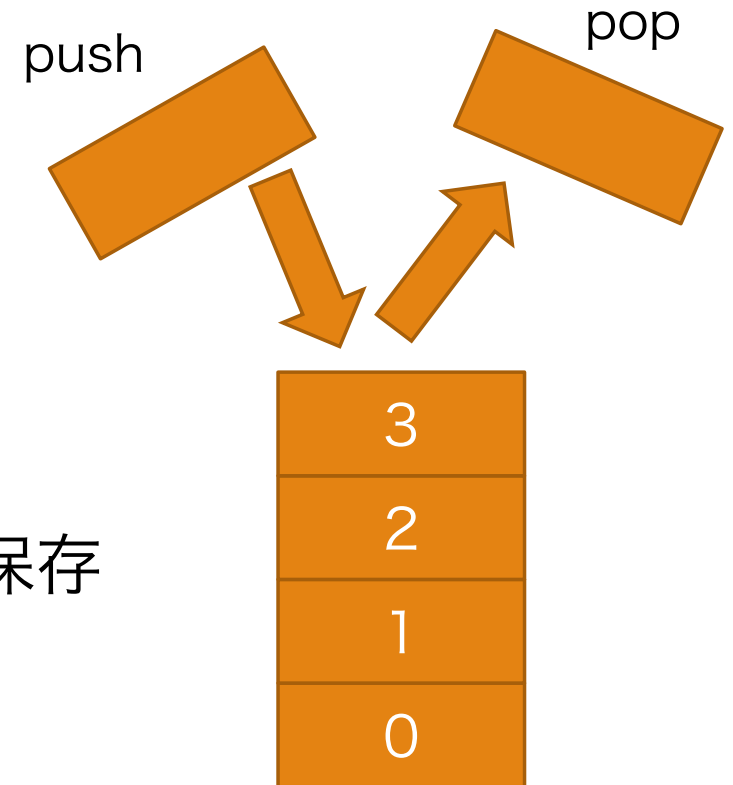


pushMatrix()

座標軸の状態を保存

popMatrix()

座標軸の状態を元に戻す



# 座標変換

---

座標変換

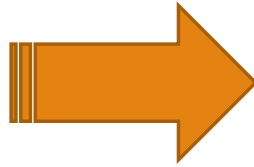
座標軸を移動させる

translate

rotate

scale

座標軸

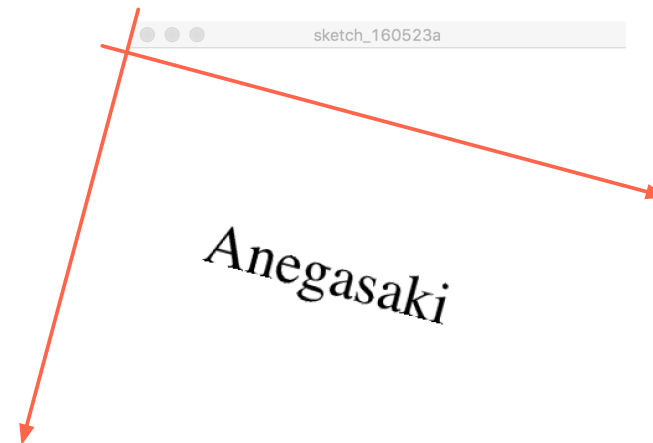


図形や文字列を表示  
する時の基準

```
text("Anegasaki", 100, height/2);
```

```
座標軸を15度回転  
rotate(radians(15))  
rotate(PI/12);
```

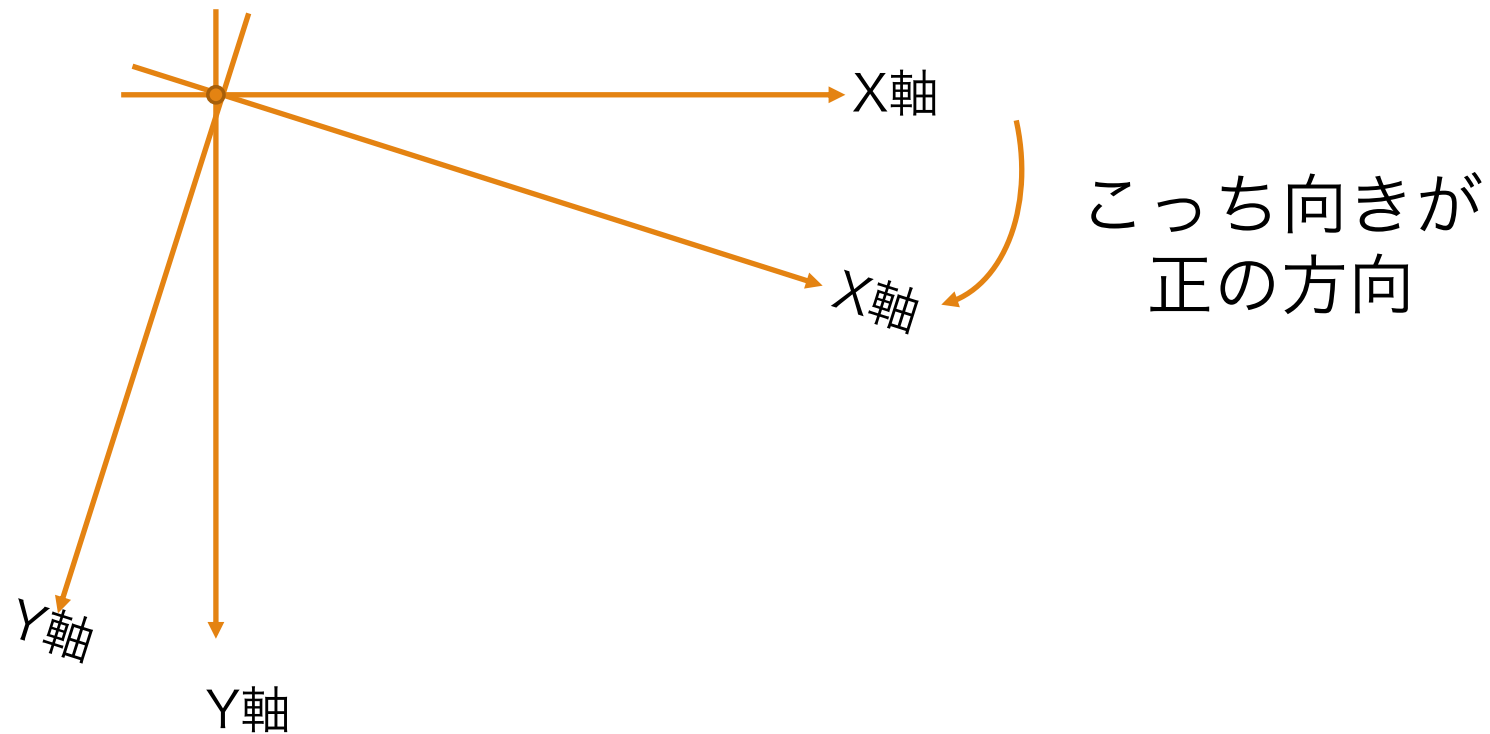
デフォルトの座標軸





# 回転角度の正の方向

---

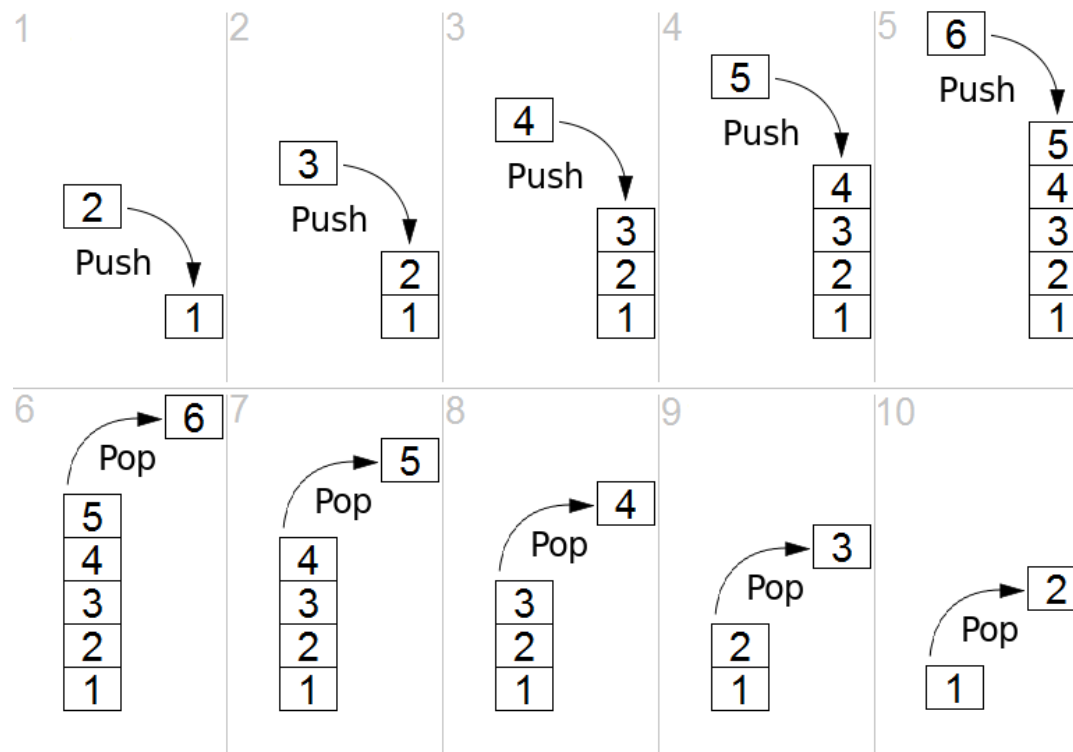


| 関数名など       | 説明                                           |
|-------------|----------------------------------------------|
| translate   | 座標軸の原点を移動する                                  |
| rotate      | 現在の原点の場所を中心に座標軸を回転させる                        |
| scale       | 座標軸の目盛の基準を拡大・縮小させる                           |
| resetMatrix | 座標軸の位置を初期状態に戻す<br>ウインドの左上が原点、X軸は水平方向、Y軸は垂直方向 |
| pushMatrix  | 現在の座標軸の状態をスタックに積む                            |
| popMatrix   | 座標軸の状態をスタックの先頭の状態にする、<br>スタックからは取り除かれる       |



# スタック

---



| 関数名など    | 説明                        |
|----------|---------------------------|
| hour()   | 現在の時間の時（0～23の整数）を返す関数。    |
| minute() | 現在の時間の分(0～59の整数)を返す関数。    |
| second() | 現在の時間の秒(0～59の整数)を返す関数。    |
| year()   | 現在の年を返す関数。                |
| month()  | 現在の月(1～12の整数)を返す関数。       |
| day()    | 現在の日(1～31の整数)を返す関数。       |
| millis() | プログラムを実行してからの時間をミリ秒単位で返す。 |

# 変数の有効範囲

---

大域変数  
(グローバル変数)

局所変数  
(ローカル変数)

区別  
どこで変数が  
宣言されたか？

違い  
どの範囲で使うことが  
出来るのか？

# 大域変数 (グローバル変数)

---

プログラムの先頭部分

変数宣言した後は、基本的にはプログラム中のどこでもOK

大域変数の使いすぎには注意

どこで変数が  
宣言されたか？

どの範囲で使うことが  
出来るのか？

# 局所変数（ローカル変数）

---

ブロックの中で変数宣言

宣言したブロック内で利用可能

```
for(int i=0;i < 10;i++){  
    // 変数iはこの中で有効  
}
```

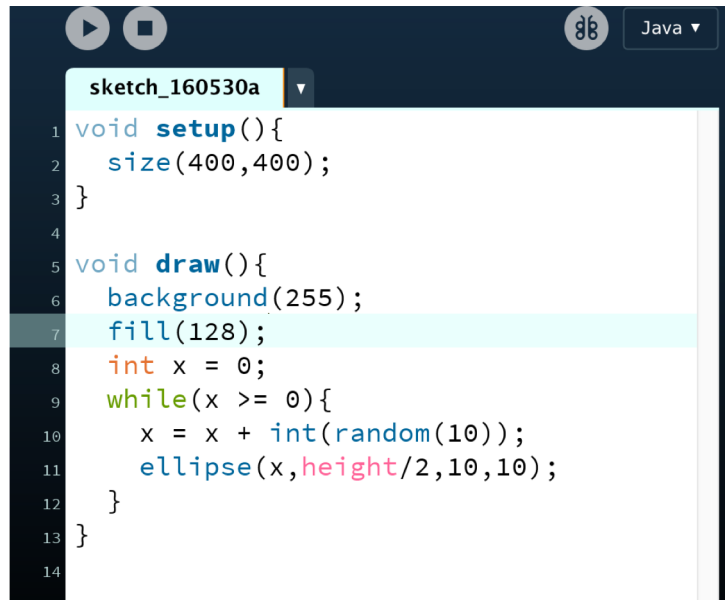
```
void draw(){  
    background(255);  
    fill(128);  
    int x = 0;  
    while(x < width){  
        x = x + int(random(10));  
        ellipse(x,height/2,10,10);  
    }  
}
```

大域変数と同じ名前の局所変数を  
宣言するとどうなるか？

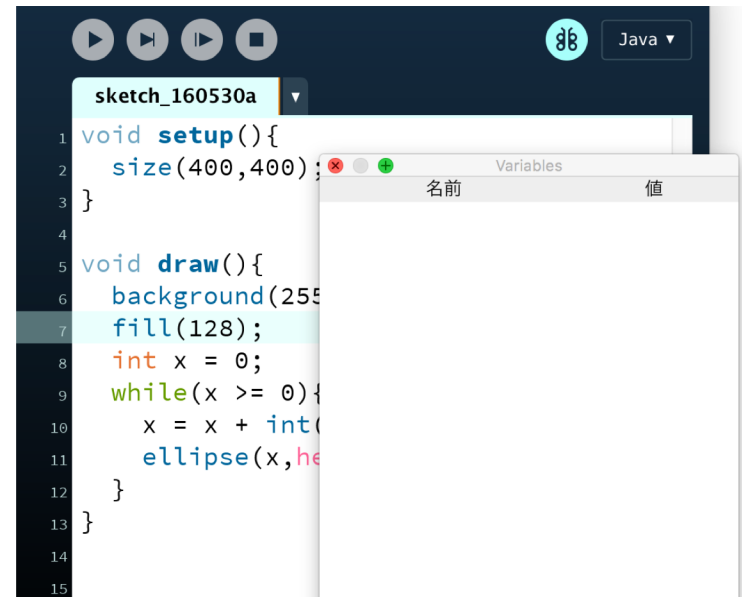
グローバル変数と局所変数は同時に存在をする  
が、宣言したブロック内部では、局所変数にし  
かアクセス出来ない。



## 通常のウィンドウ



## デバッグ状態の ウィンドウ



# 具体的には

---

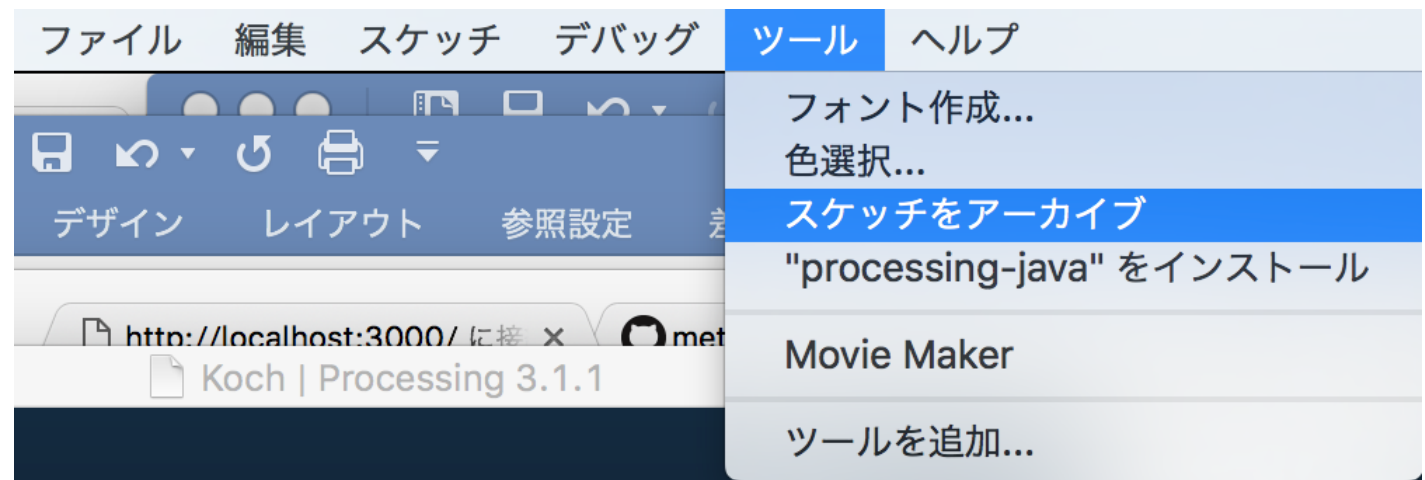
1. ステップ実行：プログラムを1ステップ毎に実行できる機能
  - stepボタンを押すと、1ステップ毎に実行。(左端に実行行にマークがつく)
2. ブレークポイント:実行を停止したい行を設定できる機能
  - ブレークポイントの設定：設定したい行にカーソルを合わせて、
  - Runボタンを押すと、ブレークポイントを設定した行でプログラムが一時停止する。
3. インспекター：変数名、値を確認できる機能
  - 変数名と値を確認できる

# クイズの提出に関して

---

キャリアポートフォリオ上で解答して下さい。

「スケッチをアーカイブ」で保存されるファイルをアップロードして下さい。



# 今日やった内容

---

座標変換

変数の有効範囲

関数その1

デバッグ



今日の内容はちょっと難しいかも？

# 授業時に配布した資料

---

<http://www.sato-lab.jp/imfu/index.html>

にあります。