2019 プログラミング入門 9回　　理工学部・情報科学科

座標変換と配列

座標変換の命令(メソッド)

通常は、左上が原点(0, 0)で右にx軸、下にy軸だが、
座標変換のメソッドで原点の位置や向きを変更できる。

translate(x軸上の移動量, y軸上の移動量);　　座標の移動
rotate(角度);　　座標の回転（角度の単位はラジアン）
　　radians メソッドで度をラジアンに単位の変換
scale(倍率);　　倍率の変更
pushMatrix();　　座標の保存
popMatrix();　　座標の復帰

ラジアンは弧度法での
角度の単位
度数法の360°は、
2π ラジアン

1. 平行移動 translate

新たにウインドウを開く。
([ファイル]メニューから[新規])

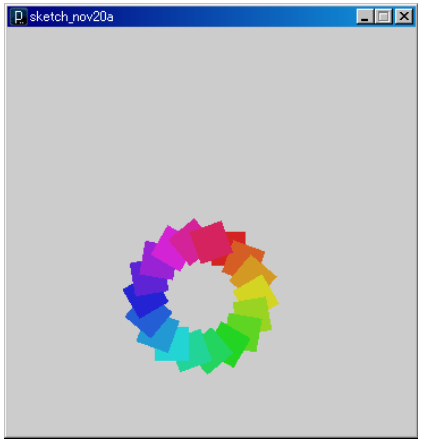
```
size(360, 360);
colorMode(HSB, 359, 99, 99);
noStroke();

for(int i = 0; i < 360; i += 20){
  fill(i, 99, 99);
  rect(0, 0, 10, 10);
  translate(20, 0);
}
```

rect の座標は　(0,0) で
変数を含んでいない
ところに注目

2. 回転 rotate

1. に太字部分を追加や修正(左に[ )

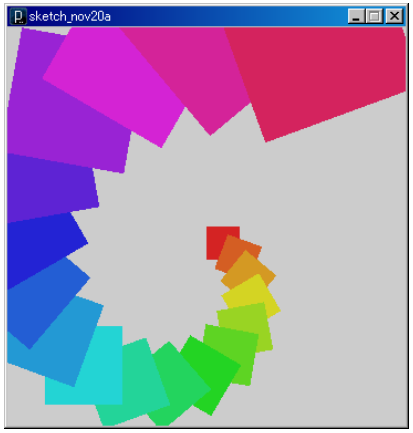
```
size(360, 360);
colorMode(HSB, 359, 99, 99);
noStroke();

translate(180, 180);
for(int i = 0; i < 360; i += 20){
  fill(i, 99, 99);
  rect(0, 0, 30, 30);
  rotate(radians(20));
  translate(20, 0);
}
```



3. 拡大・縮小 scale

2. に太字部分を追加(左に[ )

```
size(360, 360);
colorMode(HSB, 359, 99, 99);
noStroke();

translate(180, 180);
for(int i = 0; i < 360; i += 20){
  fill(i, 99, 99);
  rect(0, 0, 30, 30);
  rotate(radians(20));
  translate(20, 0);
scale(1.1);
}
```

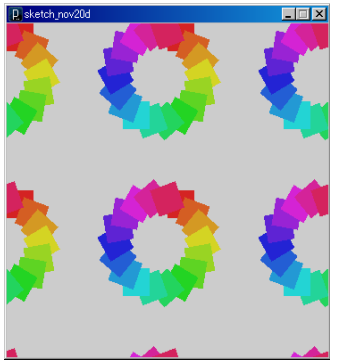


4. 座標の保存・復元 pushMatrix, popMatrix

新たにウインドウを開く。([ファイル]メニューから[新規])

```
size(360, 360);
colorMode(HSB, 359, 99, 99);
noStroke();
for (int y = 0; y <= 360; y += 180) {
  for (int x = 0; x <= 360; x += 180) {
    pushMatrix();
    translate(x, y);
    for (int i = 0; i < 360; i += 20) {
      fill(i, 99, 99);
      rect(0, 0, 30, 30);
      rotate(radians(20));
      translate(20, 0);
    }
    popMatrix();
  }
}
```

繰り返しの最初で座標を保
存しておくと、
いくら変更しても元の
座標系に戻せる。



配列

データの格納された配列を用意する。（配列の宣言と初期化）

型 [] 配列の変数の名前 = { データ };

配列の個々のデータ（要素）へのアクセス

配列の変数の名前[番号]　番号のことを添え字またはインデックスとよぶ。先頭は0番目

配列の要素数は、 配列の変数の名前 . length　で取得できる。

5. 配列　複数の同じ種類のデータをまとめて扱う

新たにウインドウを開く。([ファイル]メニューから[新規])

整数(int)の配列で名前はaをつくる。データは1, 7, 3, 2とする。

個々のデータをprintlnで表示する。

コンソールで結果を確認する。

```
int [] a = { 1, 7, 3, 2 };

println(a[0]);
println(a[1]);
println(a[2]);
println(a[3]);
```

型 [] 名前 ={ データ };

整数を扱う名前が a の配列をつくり、データで初期化

名前 [番号] の要素の値を表示

名前が a の配列 の先頭 (0 番目) の値を表示

繰り返し(for文)を使って表示を簡潔にする。以下を追加。

```
for(int i = 0; i < a.length; i++) {
  println(a[i]);
}
```

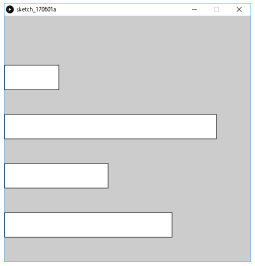
6. 配列を使ってデータに対応した大きさの複数の長方形を棒グラフ風に描く

新たにウインドウを開く。([ファイル]メニューから[新規])

```
size(500, 500);
int [] data = {110, 430, 210, 340};

for(int i = 0; i < data.length; i++){
  rect(0, (i+1)*100, data[i], 50);
}
```

ループ変数 i を使って y 座標を毎回下にずらす

配列 data の要素の値の横幅



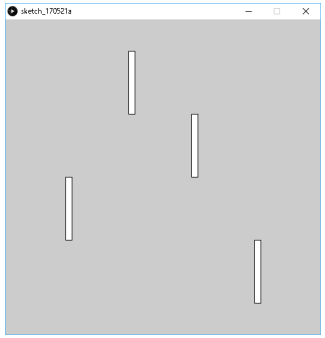
for 文の前に以下を追加すると、左に回転させて縦の棒グラフになる。

```
translate(0, 500);
rotate(radians(-90));
```


7. 配列を２つ使ってデータに対応した位置に複数の図形を描く

新たにウインドウを開く。([ファイル]メニューから[新規])

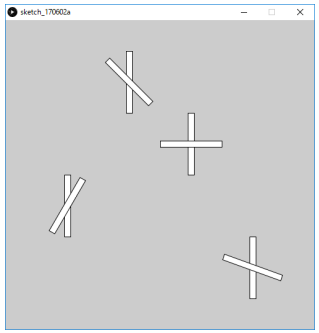
```
size(500,500);
rectMode(CENTER);
int [] x = { 100, 200, 400, 300 };
int [] y = { 300, 100, 400, 200 };

for(int i = 0; i < x.length; i++) {
  pushMatrix();
  translate(x[i],y[i]);
  rect(0, 0, 10, 100);
  popMatrix();
}
```



8. 向き（回転）のデータ用の配列と回転と描画を追加する（太字箇所:左に [）

```
  ...
int [] y = { 300, 100, 400, 200 };
int [] r = { 30, -45, 110, 90 };
for(int i = 0; i < x.length; i++) {
  pushMatrix();
  translate(x[i],y[i]);
  rect(0, 0, 10, 100);
  rotate(radians(r[i]));
  rect(0, 0, 10, 100);
  popMatrix();
}
```

回転前の図形が不要の場合はここを削除



9. アレンジ

3.4.6.7.8. どれかをアレンジ。色や形を変える。setup, drawにして変化させる等。

プログラムの提出(9週)　K's Lifeのレポート機能にて提出

提出は、3.4.6.7.8.のどれかをベースに少しでもアレンジがあるもの。

・以下をコメントとして入力（入らない場合は３つ目の提出ファイルとして追加してもよい）

今回の内容の概要、工夫した点

質問（何かあれば）・感想　難しかった。簡単だった。進め方が早い。遅い。など

・提出ファイルとして プログラム(*.pde)と実行画像(PNG または JPG)

・提出ファイルの名称にはそれぞれ学籍番号を入力

講義資料・演習資料　http://www.is.kyusan-u.ac.jp/~kamiya/pp.html