

構造を理解しよう setup と draw という命令 (メソッド) を作る

命令のしかた (命令を使う)

命令の名前 (データ, ...);

例 size(256,256);
rect(50,50,50,50);
noStroke();

命令の作り方

```
void 命令の名前() {
  文 1;
  文 2;
  文 3;
  ...
}
```

実行したい処理

実行したい処理
(設定や描画の命令を並べた)

動きを
作りたい

```
void setup() {
  最初に1度実行したい処理 (初期設定)
}
void draw() {
  繰り返し実行したい処理 (描画)
}
```

1. 動きのあるプログラム 1: マウス操作で画像を描く

mouseX, mouseY はポインタの座標の値を持つ変数: 変化する!

処理は右によせる (TABキー)

```

void setup(){
  size(256,256);
  noStroke();
  frameRate(30);
}

void draw(){
  fill(mouseX, 0, 0);
  ellipse(mouseX, mouseY, 50, 50);
}
    
```

波括弧 { } もペア 開き括弧と閉じ括弧の間に処理を書く

frameRate は画面の更新間隔の設定



2. 前回の応用

- 色の設定で mouseY を赤以外の色に使うとマウスの縦の位置でも色を変えてみよう。
- println(mouseX+", "+mouseY); を draw の中の ellipse(mouseX, mouseY, 50, 50); の次に追加して数値の変化を見てみよう。(左端が 0 で、右端が 255)
- 丸の大きさを変えてみよう

3. フェードアウト (速度に応じて大きさを変え、徐々にフェードアウトする。)

fade という命令 (メソッド) を作って使う。

1. のプログラムに **太字箇所** を追加修正。斜体は入力済み。

```

void setup(){
  size(256, 256);
  noStroke();
  frameRate(30);
}

void draw(){
  fill(mouseX, 0, 0);
  float speed = dist(mouseX, mouseY, pmouseX, pmouseY);
  ellipse(mouseX, mouseY, speed, speed);
}

void fade(){
  fill(255, 255, 255, 5);
  rect(0, 0, width, height);
}
    
```

fade の命令を使う。どう書けばよい?

fade の命令を作る。

fill の数値を変更してみよう。



pmouseX と pmouseY は、それぞれ前のフレームのマウスポインタの座標の値を持つ変数

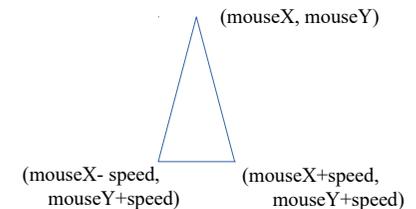
width と height は、描画ウィンドウの幅と高さの値を持つ変数

dist は距離を計算する命令 (メソッド)

float speed の部分は、変数を作る文 (次回説明する)

4. アレンジ

- 丸ではなく四角形 (rect) や他の図形に変えてみよう。
例 triangle(mouseX, mouseY, mouseX-speed, mouseY+speed, mouseX+speed, mouseY+speed);
- 最後の rect メソッドの width や height を半分にする? width/2, rect/2
- 黒くフェードアウトするには? 赤の場合は?
- 前回の絵のプログラムの size 以外をまとめた命令を作り、draw の中に追加してみよう。
例えば車の絵なら void car() { 処理 } で作り、draw の中で car(); と使う。



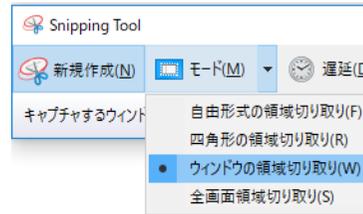
プログラムの提出(4週) K's Lifeのレポート機能にて提出

- ・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

実行画像の作り方

(frameRate(30); を frameRate(5);ぐらいにして動作を遅くしておく和良好的。)

1. Snipping Toolを起動する。
2. モードの横の三角形をクリックし、「ウィンドウの領域切り取り」に切り替える。(もしくは「四角形…」)
3. キャプチャしたいウィンドウをクリックする。(または範囲をドラッグ。)
4. ファイルメニューから名前を付けて保存する。(または、フロッピーディスクのアイコン)



5. 動きのあるプログラム 2

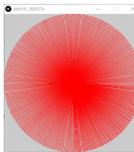
新しくウィンドウを開く ([ファイル]-[新規]).

以下を入力する。(仕組みはまだわからなくてよい。次回説明する**変数**を使っている。)

```
int ct = 0;

void setup(){
  size(360, 360);
  frameRate(30);
}

void draw(){
  ct = (ct + 1) % 360;
  stroke(255,0,0);
  line(180, 180, 180 + 180 * sin(radians(ct)), 180 + 180 * cos(radians(ct)) );
}
```



* はアスタリスク (*)
フォントによっては* のように表示される。
(ちなみに掛け算の記号)

6. fade

5. のプログラムに fade の命令を追加しよう。命令を作るところと使うところが必要。

7. 動きのあるプログラム 3 (たくさん描くには繰り返しの for 文を使う)

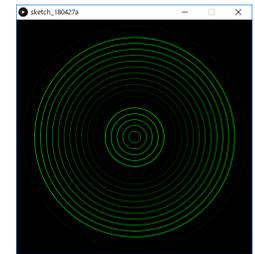
新しくウィンドウを開く ([ファイル]-[新規]).

以下を入力する。(仕組みはまだわからなくてよい。8週目に説明する**繰り返し**を使っている。)

```
int count = 0;

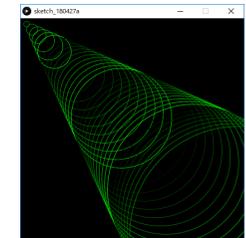
void setup() {
  size(400, 400);
  noFill();
}

void draw() {
  background(0);
  count++;
  for (int i = 0; i < 20; i++) {
    stroke(0, ((i+count) * 20) % 250, 0);
    ellipse(200, 200, 20 * i, 20 * i);
  }
}
```



for 文で複数の円を描く

i < 20 の数値を変更してみよう。



8. 応用 (直感でチャレンジ)

7. のプログラムの

- 図形を変えるには?
- 色を青にするには?
- 円の間隔を狭くするには?
- 円の中心の座標もずらすには?

よく使う命令 (メソッド)

- 画面のサイズを設定
- 楕円を描く
- 四角形を描く
- 線の色を決める
- 線の太さ
- 線を描かないようにする
- 塗りつぶしの色を決める
- 塗りつぶさないようにする

- size(横のピクセル数, 縦のピクセル数);
- ellipse(中心の x, 中心の y, 横直径, 縦直径);
- rect(左上の x, 左上の y, 幅, 高さ);
- stroke(Rの値, Gの値, Bの値);
- stroke(Rの値, Gの値, Bの値, 透明度);
- strokeWeight(太さの値);
- noStroke();
- fill(Rの値, Gの値, Bの値);
- fill(Rの値, Gの値, Bの値, 透明度);
- noFill();