

# プログラミング演習I

## 第5回: 反応1

担当: 小久保温

# アニメーション

---

- ▶ 絵を動かす
  - ▶ パラパラまんがの方法を使っている ※「仮現運動」
- ▶ フレーム frame
  - ▶ パラパラまんがの1枚1枚のことをフレームという
  - ▶ アーケードゲームなどは毎秒60フレーム
- ▶ フレームごとに絵を描く(何かさせる)には **draw()** を使う

※draw 描画

```
void draw() {  
  // フレームカウントをコンソールに表示  
  println("I'm drawing");  
  println(frameCount);  
}
```

※frameCount

現在、何番目のフレームかが入っている変数

- ▶ println(内容) コンソールに文字を書く

# setup() と draw()

---

- ▶ **setup()**

- ▶ プログラムの起動時に1回だけ実行する

- ▶ **draw()**

- ▶ 毎フレーム実行する

```
void setup() {  
    println("I'm starting");  
}  
  
void draw() {  
    println("I'm running");  
}
```

# フツのProcessingプログラム

---

## ▶ Processingのプログラムは以下のように書く

- ▶ 変数の作成
  - ▶ プログラム全体で使うものだけ
- ▶ setup()
- ▶ draw()

## ▶ 実行される順序

- ▶ ①、②、③、③、③、③...

変数の作成

①

```
void setup() {  
  起動時に実行する内容  
}
```

②

```
void draw() {  
  毎フレーム実行する内容  
}
```

③

## 例 ex5\_3 あまりいい例ではないが...

---

### ▶ ①変数の作成

- ▶ x、y、diameterを用意

### ▶ ②setup()

- ▶ ウィンドウの大きさと塗り  
の色を設定

### ▶ ③draw()

- ▶ 背景の色と円を描く

```
int x = 280;  
int y = -100;  
int diameter = 380;
```

```
void setup() {  
  size(480, 120);  
  fill(102);  
}
```

```
void draw() {  
  background(204);  
  ellipse(x, y, diameter, diameter);  
}
```

# マウスの現在位置

---

- ▶ マウスの現在位置が入っている変数
  - ▶ **mouseX**、**mouseY** ※大文字小文字に注意
- ▶ ex5\_4

```
void setup() {  
  size(480, 120);  
  fill(0, 102); // 黒、半透明で塗る  
  noStroke(); // 枠線を引かない  
}  
  
void draw() {  
  // マウスの位置に直径9の円を描く  
  ellipse(mouseX, mouseY, 9, 9);  
}
```

# 画面をクリア

---

- ▶ これまで描いてきたものを全部クリアするには？
  - ▶ 答え: background()を使う
- ▶ background(色)
  - ▶ 全面塗りつぶし
- ▶ ex5\_5

```
void setup() {  
  size(480, 120);  
  fill(0, 102);  
  noStroke();  
}  
  
void draw() {  
  background(204); // 灰色で画面をクリア  
  ellipse(mouseX, mouseY, 9, 9);  
}
```

## つないで描く

---

- ▶ 今までの描き方だと、線がぶっち切れる
  - ▶ 線をつないでいないから
- ▶ 線をつなぐには...
- ▶ 1つ前のフレームと、現在フレームのマウスの位置を結ぶ
- ▶ 1つ前のフレームのマウスの位置
  - ▶ `pmouseX`、`pmouseY` ※`pre` 前
- ▶ `ex5_6`

```
void setup() {
  size(480, 120);
  strokeWeight(4);
  stroke(0, 102);
}

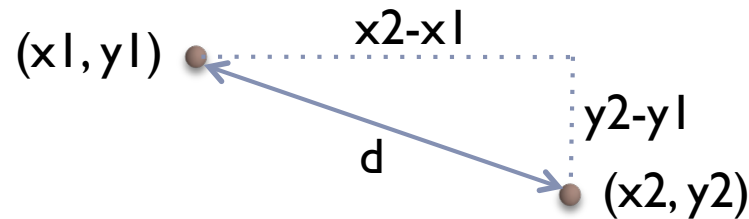
void draw() {
  line(mouseX, mouseY, pmouseX, pmouseY);
}
```



# 長さ

---

- ▶ 2つの点の間の長さは「三平方の定理」で計算できる



$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

- ▶ これを全部やってくれるのが `dist()`
  - ▶ `dist(x1, y1, x2, y2)` これだけでOK

## 使用例 ex5\_7

---

### ▶ 移動距離を太さに

```
void setup() {  
    size(480, 120);  
    stroke(0, 102);  
}  
  
void draw() {  
    // 移動距離を太にする  
    float weight = dist(mouseX, mouseY, pmouseX, pmouseY);  
    strokeWeight(weight);  
    line(mouseX, mouseY, pmouseX, pmouseY);  
}
```

# 小数を使う

---

- ▶ 小数の型 **float**
- ▶ 小数型の変数 **weight** の宣言

```
float weight;    ※weight 太さ、体重
```

- ▶ floatって？
  - ▶ 小数のことは、正式には浮動小数点数 floating point number という
  - ▶ 浮動小数点: 小数点の位置がいろいろなところにつく  
12.234   0.00134   1.5   2.0

## easing イージング: ゆっくり...

---

- ▶ 自然にもものが動いて見えるには...
  - ▶ いきなりスピードが変わるのは変
  - ▶ 最初と最後は、ゆるやかに変化した方が自然
- ▶ こういうのをプログラマ的にはイージングという

## 例 ex5\_8 ゆっくりとまる

### ▶ マウスに近くなるとゆっくり

```
float x;
// 小さくするとどんどんゆっくり
float easing = 0.01;
float diameter = 12;

void setup() {
  size(220, 120);
}

void draw() {
  float targetX = mouseX;
  x += (targetX - x) * easing;
  ellipse(x, 40, diameter, diameter);
  println(targetX + " : " + x);
}
```

- ・ 速度とは？  
「位置がどれくらい変化するか」  
 $x += x\text{方向の速度}$
- ・ マウスの位置に移動するには？  
 $x += \text{targetX} - x;$   
マウスの位置まで一気につめるぜ！
- ・ ゆっくりさせるには？  
 $x += (\text{targetX} - x) * \text{ゆっくり度};$   
ちょっとだけゆっくり…