

後ろ5列は着席禁止
3人掛けの中央は着席禁止

プログラミング入門 Processingプログラミング 第6回

九州産業大学 理工学部情報科学科
神屋郁子
(pp@is.kyusan-u.ac.jp)

時限	クラス
水1	機械 (クラス3)
水2	機械 (クラス1)
水4	電気 (B1、B2)

第6回の内容

- 前回の質問への回答
- if文 (条件分岐)
- マウスボタンを押しているときのみ描く
- 動く丸が右端なら元の位置に戻す
- if-else文
- 丸の移動の向きで図形を変える

2

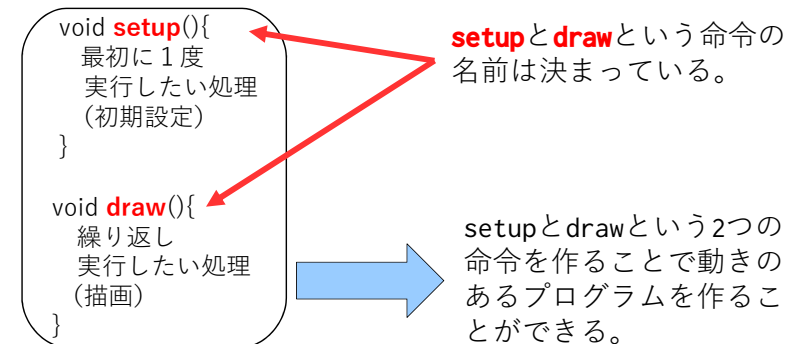
前回の質問

- for文の複数の円はこれでできてますか？
 - 変数の値をfor文で変えることで複数の円を描いていました。
for文を習ったあとでプログラムをまた読み直してみてください。
- 大きさをゆっくり変える方法が知りたい
 - floatで小数の変数を使うと0.5ずつ増やしたりできる。
- 色の配色がわからなかった
 - HSBモードは、ツールの色選択で色と数値の変化を確認しよう。
- 跳ね返りさせたい
 - 今日します！

3

復習) プログラムを構造にする

- 命令を並べて命令を作る



4

復習) 変数

- データを格納する入れ物
- 変化するデータを扱うのに使う
- processingで用意されている変数
 - mouseX, mouseY, width, height
- 自分で変数を作ることできる (変数宣言)

型 名前;

例) **int x;** 整数を扱う名前がxという変数を作る

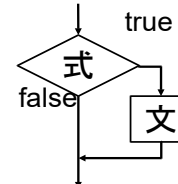


x

5

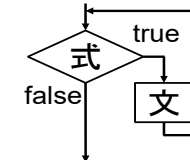
条件分岐と繰り返し

- 条件分岐



- 例: もし天気が良ければドライブに行く
- 条件に当てはまる時だけ処理を実行

- 繰り返し



- 例: 資格を取るまで勉強をする
- 条件に当てはまる間はずっと処理を続ける

6

条件分岐

- もし〇〇なら××する(条件によって行動を変える)

```
if(条件){
    処理
}
```

条件がtrue(正しい)なら文を実行

- 例: 明日、もし晴れたらドライブに行こう

```
if (tenki == HARE) {
    drive();
}
```

↑ 閉じ括弧までが一つのまとまり

開き括弧と閉じ括弧の間に条件が成り立つときの処理を書く

7

関係演算子

演算子	読み方	意味	利用例
<	小なり	～より小さい、～未満	a < b aはbより小さい
<=	小なりイコール	～以下	a <= b aはb以下
>	大なり	～より大きい	a > b aはbより大きい
>=	大なりイコール	～以上	a >= b aはb以上
==	イコールイコール	等しい	a == b aはbと等しい
!=	ノットイコール	等しくない	a != b aはbと等しくない

8

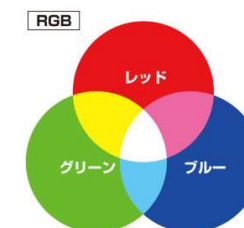
色空間

- 各色を数値の組み合わせで表現する方法または表現可能な色域
- RGBとHSB
 - RGB：赤(Red)緑(Green)青(Blue)
 - HSB：色相(Hue)彩度(Saturation) 明度(Brightness)

9

RGB

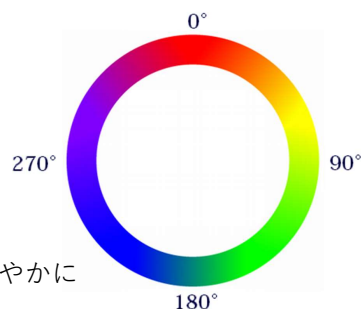
- 赤(R)緑(G)青(B)の光の三原色で色を指定する方法
 - 赤、青、緑の色の量を0～255の数値で表現
 - 赤なら(255, 0, 0)
 - 緑なら(0, 255, 0)
 - 青なら(0, 0, 255)
 - 黄色は赤と緑の両方なので(255, 255, 0)
 - すべて混ぜると白(255, 255, 255)
 - なにもないと黒(0, 0, 0)



10

HSB

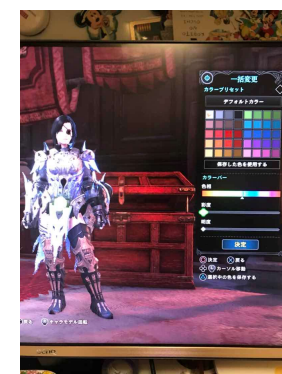
- 色相(Hue)彩度(Saturation)明度(Brightness)を数値で表現
- 色相：色合い
 - この値を変えると色合いが変化する
 - 赤は0
 - 緑は120
 - 青は240
- 彩度：鮮やかさ
 - 0～99(%) 白から始まり値が増えるほど鮮やかに
- 明度：明るさ
 - 0～99(%) 黒から始まり値が増えるほど明るく



11

こんなところで使われています。

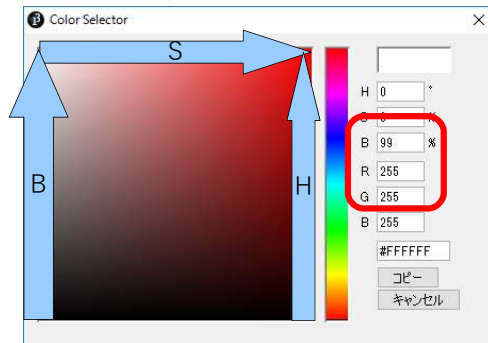
- モンスターハンターワールドの防具の着彩設定



12

HSBモード

- setupのメソッドに colorMode(HSB, 359, 99, 99); を追加し、drawメソッドの中で色を指定する



- 色相(Hue)
- 彩度(Saturation)
- 明度(Brightness)

前々回のプログラムの改良版

- 前々回
 - マウスポインタの位置に丸を描く
- ボタンを押しているときのみ描くには？
 - **もしボタンが押されてたら、丸を描く**
 - **ボタンが押されてたら**の条件はどう書く？
 - **丸を描く**処理はどう書く？

14

作業

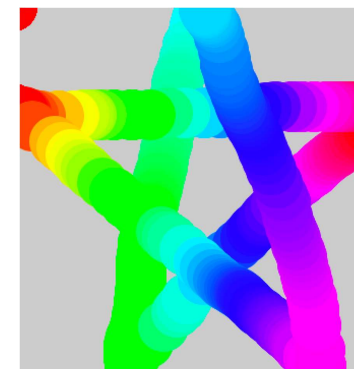
1. マウスポインタの位置に丸を描く (復習)
2. マウスのボタンが押されている時だけ丸を描く
3. 左右のボタンで色を変える
4. xが100を超えたらxを0にする

15

```
void setup(){
  size(360,360);
  colorMode(HSB,359,99,99);
  noStroke();
  frameRate(30);
}

void draw(){
  fill(mouseX,99,99);
  ellipse(mouseX,mouseY,50,50);
}
```

演習1



14

もし、マウスのボタンが押されているなら
色を決めて描く

演習2
(drawメソッドのみ)

```
void draw(){
  println(mousePressed);
  if(mousePressed==true){
    fill(mouseX,99,99);
    ellipse(mouseX,mouseY,50,50);
  }
}
```

mousePressedは
Processingが持つシ
ステム変数
マウスのボタンが押さ
れているかどうかを格
納されている変数
true(真)かfalse(偽)か
の値を持つ

```
void draw(){
  println(mousePressed+" "+mouseButton);
  if(mousePressed==true){
    if(mouseButton == LEFT){
      fill(mouseX,99,99);
    }
    if(mouseButton == RIGHT){
      fill(0,99,99);
    }
    ellipse(mouseX,mouseY,50,50);
  }
}
```

演習3
(drawメソッドのみ)

mouseButtonは
Processingが持つシ
ステム変数
どのボタンが押されて
いるかを格納する変数
(LEFT,RIGHT,CENTER)

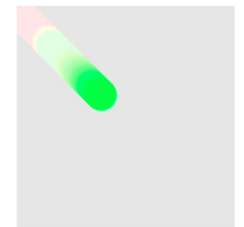
```
int x = 0;
```

演習4

```
void draw() {
  x++;
  if(x > 100){
    x = 0;
  }
  println(x);
}
```

前回のプログラムの改良

- 前回
 - ボール (丸) が左上から右下へ移動する
 - 右下まで行ったら変化なしで面白くない!
- 画面外にでたらもとの位置に戻すには?
 - **もし画面外に出たら、座標をもとの位置に**
 - **画面外に出たら**の条件はどう書く?
 - **座標をもとの位置**に戻す処理はどう書く?
- 画面外に出ないように跳ね返すには?



作業

- 右下に移動する図形を描く
- 速度をもっと速くする
- 画面外に出たら左上に戻る
- 場所によって形を変える (if-else文)
- 画面の端で跳ね返るようにする
- マウスをクリックした位置に移動する
- 移動方向 (左右) で色を変える

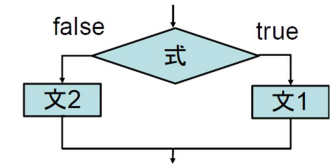
21

if-else文

- もし〇〇なら××する、そうでなければ△△する

```
if(式){  
  文1  
} else {  
  文2  
}
```

式がtrueなら文1を実行
falseなら文2を実行



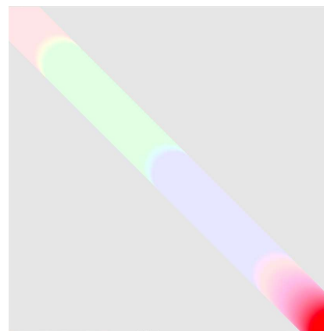
- 例: もし晴れたらドライブに、そうでなければ本を読もう
- ```
if (tenki == HARE) {
 drive();
} else {
 read();
}
```

22

```
float x,y;
```

演習5  
(前半)  
右下に移動する図形

```
void setup(){
 size(360,360);
 colorMode(HSB,359,99,99);
 noStroke();
 x = 0;
 y = 0;
}
```

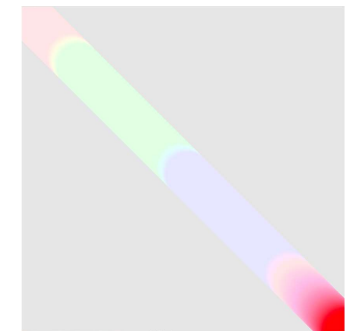


```
void draw(){
```

```
 fade();
 x += 0.5;
 y += 0.5;
 fill(x,99,99);
 ellipse(x,y,50,50);
```

```
}
void fade(){
 fill(0,0,99,5);
 rect(0,0,width,height);
}
```

演習5  
(後半)  
右下に移動する図形



```

void draw(){
 fade();
 x += 0.5;
 y += 0.5;
 fill(x,99,99);
 ellipse(x,y,50,50);
}

```

演習6  
速度を速くする  
(drawメソッドのみ)

```

void draw(){
 fade();
 x += 0.5;
 y += 0.5;
 if(x > 360){
 x = 0;
 y = 0;
 }
 fill(x,300,99,99);
 ellipse(x,y,50,50);
}

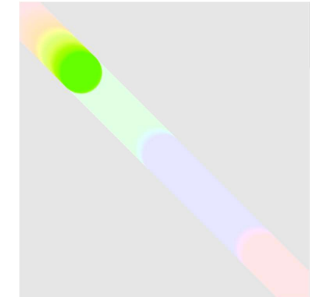
```

演習7  
画面外に移動したら  
元に戻す  
(drawメソッドのみ)

```

if(x > 360){
 x = 0;
 y = 0;
}

```

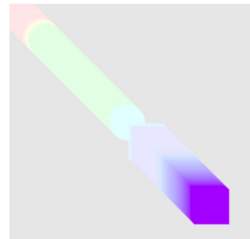


```

void draw(){
 fade();
 x += 0.5;
 y += 0.5;
 if(x > 360){
 x = 0;
 y = 0;
 }
 fill(x,300,99,99);
 if(y < 180 または height/2) { // 上半分
 ellipse(x,y,50,50); //図形1の描画
 } else {
 rect(x,y,50,50); //図形2の描画
 }
}

```

演習8  
場所に応じて形を変更  
(drawメソッドのみ)



```

float x,y,vx,vy;

void setup(){
 size(360,360);
 colorMode(HSB,359,99,99);
 noStroke();
 x = 0;
 y = 0;
 vx = 4.5;
 vy = 1.5;
}

```

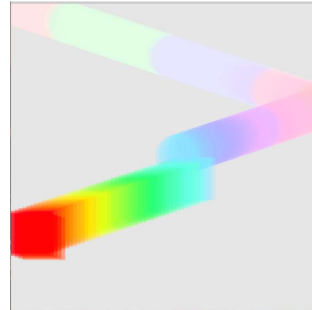
演習9  
画面の端で跳ね返る  
(前半)

```

void draw(){
 fade();
 x += vx;
 y += vy;
 if(x > 360){
 vx = -vx;
 }
 if(x < 0){
 vx = -vx;
 }
 if(y > 360){
 vy = -vy;
 }
 if(y < 0){
 vy = -vy;
 }
 fill(x,99,99);
}

```

演習9  
画面の端で跳ね返る  
(後半)



```

void draw(){
 fade();
 if(mousePressed==true) {
 x = mouseX;
 y = mouseY;
 vx = random(-3,3);
 vy = random(-3,3);
 }
 fill(x,99,99);
}

```

演習9の座標関係の処理

**if(mousePressed==true)** {

```

 x = mouseX;
 y = mouseY;
 vx = random(-3,3);
 vy = random(-3,3);

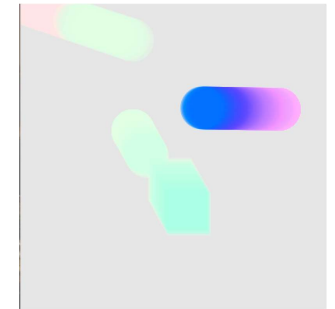
```

```

}
fill(x,99,99);

```

演習10  
マウスの位置に移動  
(後半)



```

void draw(){
 fade();
 if(vx > 0){
 fill(120,99,99,50);
 } else {
 fill(240,99,99,50);
 }
}

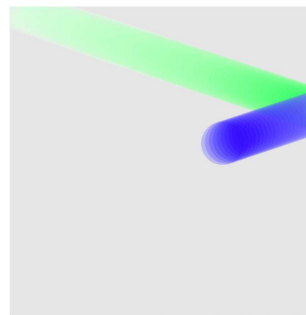
```

演習11  
移動方向で色変更  
(後半)

演習9の座標関係の処理

演習10のボタンが押された時の処理


演習8の図形を描く処理



## プログラムの提出 (演習点)

- if-else文を用いたプログラムを作ってみよう！
- K's Lifeのレポート機能から
- 締め切りは6日後の23:59
  - それ以降も受け取るが**減点**する
  - もっと改良したくてもこの時間に一度提出しよう
  - (削除して提出しなおし可能)
- 評価はK's Life上ではしない





## 提出する内容

- 以下をコメントとして入力
  - 今回の内容の概要
  - (改良した点があれば) 工夫した点
  - 質問 (何かあれば) ・感想 (簡単だった・難しかったなど)
  - 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
- 提出ファイルとして以下の2つ
  - プログラム (\*.pde)
  - 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