

## 2017 年度情報メディア基盤ユニット 6月6日講義内で作ったサンプルプログラム

### その1

```
// 組み込み関数 dist の利用例
void setup() {
  size(400, 400);
}

void draw() {
  background(255);
  // 2点(width/2,height/2) (mouseX,mouseY)の間の距離を求める
  float r = dist(width/2, height/2, mouseX, mouseY);

  stroke(0);
  fill(255, 10, 10);
  ellipse(width/2, height/2, 2*r, 2*r);
}
```

### その2

```
// 組み込み関数 atan2 の利用例
void setup() {
  size(400,400);
}

void draw() {
  background(255);
  float dx = mouseX - width/2;
  float dy = mouseY - height/2;
  float angle = atan2(dy,dx);
  drawArrow(width/2, height/2, angle, color(10, 255, 10));
}

void drawArrow(float x, float y, float angle, color material) {
  float length = 50;
  rectMode(CENTER);
  stroke(material);

  pushMatrix();
  translate(x, y);
  rotate(angle);
  fill(material);
  rect(0, 0, 0.75*length, 0.35*length);
  triangle(0.5*length, 0,
    0.25*length, 0.3*length,
    0.25*length, -0.3*length);
  popMatrix();
}
```

### その3

```
// 組み込み関数 atan2 の利用例
void setup() {
```

```

    size(400, 400);
}

void draw() {
    background(255);

    // 2点(mouseX,mouseY)、(width/2,height/2)を結ぶ線分と X 軸のなす角度
    float dx = mouseX - width/2;
    float dy = mouseY - height/2;
    float angle = atan2(dy, dx);

    rectMode(CENTER);
    translate(width/2, height/2);
    rotate(angle);
    fill(255, 10, 10);
    rect(0, 0, 30, 30);
}

```

#### その4

// 戻り値なし、引数無し関数宣言例

```

float xPos;
float yPos;

void setup() {
    size(400, 400);
    background(255);
}

/*
void draw() {
    noStroke();
    fill(255, 255, 255, 100);
    rect(0, 0, width, height);
    stroke(0);
    fill(255, 10, 10);
    yPos = yPos+1;
    xPos = width/2 + 0.40 * width *sin(radians(millis()/10.0));
    ellipse(xPos, yPos, 20, 20);
}
*/

void draw() {
    fadeToWhite();
    stroke(0);
    fill(255, 10, 10);
    move();
    ellipse(xPos, yPos, 20, 20);
}

void fadeToWhite() {
    noStroke();
    fill(255, 255, 255, 100);
    rect(0, 0, width, height);
}

```

```
void move() {
  yPos = yPos+1;
  xPos = width/2 + 0.40 * width *sin(radians(millis()/10.0));
}
```

## その5

// 戻り値なし、引数ありの関数宣言例

```
float xPos;
float yPos;
```

```
void setup() {
  size(400, 400);
  background(255);
}
```

```
void draw() {
  fadeToWhite(100);
  stroke(0);
  fill(255, 10, 10);
  move();
  ellipse(xPos, yPos, 20, 20);
}
```

```
void fadeToWhite(float alpha) {
  noStroke();
  fill(255, 255, 255, alpha);
  rect(0, 0, width, height);
}
```

```
void move() {
  yPos = yPos+1;
  xPos = width/2 + 0.40 * width *sin(radians(millis()/10.0));
}
```

## その6

// マウスの位置に向かって移動する円

```
float xPos;
float yPos;
```

```
void setup(){
  size(600,600);
  xPos = width/2;
  yPos = height/2;
}
```

```
void draw(){
  background(255);

  xPos = 0.95*xPos + 0.05*mouseX;
  yPos = 0.95*yPos + 0.05*mouseY;
  fill(255,10,10);
  ellipse(xPos,yPos,20,20);
}
```

```
}
```

### その7

```
// 戻り値無し、引数なしの関数宣言の例
```

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

```
void draw() {  
  background(255);  
  stroke(255, 10, 10);  
  drawRect();  
  /*  
  for(int x=0;x < 60;x++){  
    line(mouseX+x,mouseY,mouseX+x,mouseY+40);  
  }*/  
}
```

```
// 引数無し関数宣言例、ちょっと汎用性に欠ける
```

```
// 長方形を描く関数
```

```
void drawRect() {  
  for (int x=0; x < 60; x++) {  
    line(mouseX+x, mouseY, mouseX+x, mouseY+40);  
  }  
}
```

### その8

```
// 戻り値無し、引数ありの関数宣言の例
```

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

```
void draw() {  
  background(255); //<> // //<> //  
  stroke(255, 10, 10);  
  drawRect(0,0,30,40);  
  drawRect(mouseX,mouseY,60,40);  
}
```

### その9

```
// 引数ありの関数宣言例
```

```
// 長方形を描く関数
```

```
void drawRect(float x0,float y0,float w,float h) {  
  for (int x=0; x < w; x++) {  
    line(x0+x, y0, x0+x, y0+h);  
  }  
}
```

```
// 戻り値なし、引数無し関数宣言例
```

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

```

void draw(){
  background(255);
  translate(mouseX,mouseY);
  drawPackMan();
  /*
  pushMatrix();
  float angle=PI/6;
  fill(255,10,10);
  arc(0, 0, 40, 40, angle, 2*PI-angle);
  rotate(angle);
  line(0, 0, 20, 0);
  rotate(-2*angle);
  line(0, 0, 20, 0);
  popMatrix();
  */
}

```

```

void drawPackMan(){
  pushMatrix();
  float angle=PI/6;
  fill(255,10,10);
  arc(0, 0, 40, 40, angle, 2*PI-angle);
  rotate(angle);
  line(0, 0, 20, 0);
  rotate(-2*angle);
  line(0, 0, 20, 0);
  popMatrix();
}

```

## その10

// 引数なし、戻り値ありの関数宣言例

```

String today(){
  String msg = year() + "/" + month() + "/" + day();
  return msg;
}

```

```

String currentTime(){
  String time = hour() + ":" + minute() + ":" + second();
  return time;
}

```

```

int minutes(){
  int total;
  total = 60*hour() + minute();
  return total;
}

```

```

void setup(){
}

```

```

void draw(){
  String msg = today(); //<>//
  println(frameCount+" "+msg);
}

```

```
println(currentTime());
println(minutes());
}
```

## その 11

// 引数あり、戻り値ありの関数宣言例  
// (x,y)を中心とする半径 20 の円と 2 点(xTip,yTip),(xTip,0)を結ぶ線分が交差するかどうかを判定する関数

```
boolean doesCollide(int x, int y, int xTip, int yTip) {
    float d = dist(x, y, xTip, yTip);
    if (d <= 20) {
        return true;
    } else if ((abs(x - xTip) <= 20) && (y <= yTip)) {
        return true;
    } else {
        return false;
    }
}
```

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

```
void draw() {
    background(255);
    stroke(0);
    if (doesCollide(mouseX, mouseY, width/2, height/3)) {
        fill(255, 10, 10);
    } else {
        fill(100);
    }
    line(width/2, 0, width/2, height/3);
    ellipse(mouseX, mouseY, 2*20, 2*20);
}
```

## その 12

// コールバック関数 mouseClicked の利用例  
color fg;

```
void setup() {
    size(600, 400);
    colorMode(HSB, 359, 99, 99);
    fg = color(0, 99, 99);
}
```

```
void draw() {
    background(0, 0, 99);
    fill(fg);
    ellipse(width/2, height/2, 100, 100);
}
```

// マウスがクリックされたら呼び出されるコールバック関数  
void mouseClicked() {

```
println("mouse clicked");
fg = color(random(360), 99, 99);
}
```

### その 13

```
// コールバック関数 mouseClicked の利用例
color fg;
```

```
void setup(){
  size(600,400);
  colorMode(HSB,359,99,99);
  fg = color(0,99,99);
}
```

```
void draw(){
  background(0,0,99);
  fill(fg);
  ellipse(width/2,height/2,100,100);
}
```

```
// マウスがクリックされたら呼び出されるコールバック関数
```

```
void mouseClicked(){
  println("mouse clicked");

  if(mouseButton == LEFT){
    // 左ボタンがクリックされたとき
    fg = color(random(360),99,99);
  }else if(mouseButton == RIGHT){
    // 右ボタンがクリックされたとき
    fg = color(0,99,99);
  }
}
```