

担当：モトムラケンタ

## Processing（プロセッシング）によるヴィジュアル表現の基礎（の基礎）

---

この練習は、インストールした Processing で行うことを前提としています。  
Processing のスケッチ（作品）を公開したり、共有したりできるサイト「OpenProcessing」  
で作業を行う場合は下記の例のように少し書き方が変わるので注意します。

```
function setup() {  
    createCanvas(windowWidth, windowHeight);  
    background(100);  
}  
function draw() {  
    ellipse(mouseX, mouseY, 20, 20);  
}
```

以下の練習を順に、スケッチ(Sketch)にコピー&ペストで実行して結果を確認しながら学習を進めてください。

練習 1（P01）：基礎の基礎

//コメント：下記の URL 参照

[//https://processing.org/reference/](https://processing.org/reference/)

//display window width and height in units of pixels（画面サイズ）

size(600, 600);

//width of the stroke（線の幅）

strokeWeight(6);

//2D Primitives（平面の基本形：点・線・三角形・四角形・円）

point(width/2, height/2);//点

strokeWeight(2);

line(0, 0, width, height);//線

triangle(30, 75, 58, 20, 86, 75);//三角形

rect(530, 530, 55, 55);//四角形

rect(430, 430, 65, 65, 8);//四角形（角丸）

quad(138, 131, 186, 120, 169, 163, 130, 176); //四角形（4点で）

//disables filling geometry（色を塗らない）

noFill();

ellipse(width/2, height/2, 50, 50);//円

~~~~~

練習 2 (P02) : 図形を描く

[//https://processing.org/reference/](https://processing.org/reference/)

[//define initial enviroment properties such as screen size and to load media such as images and fonts as the program starts \(setup\(\)で初期設定\)](#)

```
void setup() {
```

```
  size(600, 600);
```

```
  background(255);
```

```
  //background(255, 204, 0);
```

```
}
```

[//continuously executes the lines of code contained inside its block until the program is stopped or noLoop\(\) is called \(draw\(\)は止めない限り処理を繰り返す\)](#)

```
void draw() {
```

```
  //width of the stroke (線の幅)
```

```
  strokeWeight(6);
```

```
  //2D Primitives (平面の基本形)
```

```
  point(width/2, height/2);
```

```
  strokeWeight(2);
```

```
  line(0, 0, width, height);
```

```
  triangle(30, 75, 58, 20, 86, 75);
```

```
  //sets the color used to fill shapes (塗の色を設定 : R,G,B)
```

```
  /*パラメータの数 :
```

```
    1 - gray
```

```
    2 - gray, alpha
```

```
    3 - R, G, B or H, S, B
```

```
    4 - R, G, B, alpha or H, S, B, alpha
```

```
  */
```

```
  fill(204, 102, 0);
```

```
  rect(530, 530, 55, 55);
```

```
  rect(430, 430, 65, 65, 8);
```

```
  //sets the color used to draw lines and borders around shapes (線の色)
```

```
  stroke(95, 118, 180);
```

```
  quad(138, 131, 186, 120, 169, 163, 130, 176);
```

```
  //disables filling geometry (中に色を塗らない)
```

```
  noFill();
```

```

    ellipse(width/2, height/2, 50, 50);
    noLoop();//繰り返しを止める
}
//saves an image from the display window (スペースキーを押したら画像保存)
void keyPressed() {
    if ( key == ' ' ) {
        save( "P02.jpg" );
    }
}

```

~~~~~

練習 3 (P03) : 図形を描く (課題 1)

//EXERCISE

```

void setup() {
    size(800, 800);//サイズの変更可
    background(255);//背景色の変更可
}

void draw() {
    //課題 1 : ここに各自で点・線・面を書き、色面構成を下さい
    noLoop();
}

//saves an image from the display window
void keyPressed() {
    if ( key == ' ' ) {
        save( "Ex01.jpg" );
    }
}

```

~~~~~

練習 4 (P04) : インタラクティブな描画 1

//<https://processing.org/reference/>

```

void setup() {
    size(600, 600);

    //colorMode(HSB_mode, hue_max, saturation_max, brightness_max, alpha_max)
    (HSB : 色相,彩度,明度,不透明度による色の指定モード)
    colorMode(HSB, 360, 100, 100, 100);
}

```

```

    background(0);
}
//continuously executes the lines of code contained inside its block until the program is
stopped or noLoop() is called
void draw() {
    strokeWeight(8);
    //Sets the color used to draw lines and borders around shapes (線の色)
    stroke(359);
    point(width/2, height/2);
    strokeWeight(2);
    stroke(30, 60, 99, 30);
    line(width/2, height/2, mouseX, mouseY);//マウスの座標に追従する
    //noLoop(); (コメントアウトでループさせている)
}
//saves an image from the display window
void keyPressed() {
    if ( key == ' ' ) {
        save( "P04.jpg" );
    }
}

```

~~~~~

練習 5 (P05) : インタラクティブな描画 2

[//https://processing.org/reference/](https://processing.org/reference/)

```

void setup() {
    size(600, 600);
    colorMode(HSB, 360, 100, 100, 100);
    background(0);
    //マウスの座標をまずは画面中心に持っていく
    mouseX = width/2;
    mouseY = height/2;
}
void draw() {
    strokeWeight(8);
    stroke(359);
    point(width/2, height/2);

```

```

    strokeWeight(2);
    stroke(60, 60, 99, 30);
    line(width/2, height/2, mouseX, mouseY);
}
//saves an image from the display window
void keyPressed() {
    if ( key == ' ' ) {
        save( "P05.jpg" );
    }
}

```

~~~~~

練習 6 (P06) : インタラクティブな描画 3

[//https://processing.org/reference/](https://processing.org/reference/)

```

void setup() {
    size(600, 600);
    colorMode(HSB, 360, 100, 100, 100);
    background(0);
    mouseX = width/2;
    mouseY = height/2;
}
void draw() {
    strokeWeight(8);
    stroke(359);
    point(width/2, height/2);
    strokeWeight(2);
    stroke(60, 60, 99, 30);
    line(width/2, height/2, mouseX, mouseY);
    strokeWeight(1);
    stroke(120, 60, 60, 20);
    line(mouseX, 0, mouseX, height);
}
//saves an image from the display window
void keyPressed() {
    if ( key == ' ' ) {
        save( "P06.jpg" );
    }
}

```

```
}  
}
```

~~~~~  
練習 7 (P07) : インタラクティブな描画 (課題 2)

[//EXERCISE](#)

```
void setup() {  
  size(800, 800);  
  colorMode(HSB, 360, 100, 100, 100);  
  background(0);  
  mouseX = width/2;  
  mouseY = height/2;  
}  
  
void draw() {  
  //noStroke\(\); (コメントアウトで線が描かれているので // を消してみる)  
  //課題 2 : 以下を加筆修正して、マウスに追従して描かれる図形を考える  
  fill(310, 50, 50, 20);  
  ellipse(mouseX, mouseY, 30, 30);  
}  
  
//saves an image from the display window  
void keyPressed() {  
  if ( key == ' ' ) {  
    save( "Ex02.png" );  
  }  
}
```

~~~~~  
練習 8 (P08) : ランダム設定 (クリックするまで描き続ける)

[//https://processing.org/reference/](https://processing.org/reference/)

```
void setup() {  
  size(600, 600);  
  colorMode(HSB, 360, 100, 100, 100);  
  background(0);  
}  
  
void draw() {  
  //Blends the pixels in the display window according to a defined mode (色のブレンド)
```

//BLEND, ADD, SUBTRACT, DARKEST, LIGHTEST, DIFFERENCE, EXCLUSION,  
MULTIPLY, SCREEN, REPLACE

```
blendMode(ADD);  
//random()によってランダム設定  
float x = random(width); // float は整数ではない「実数」を表す変数  
float y = random(height);  
float w = random(20, 100);  
float h = random(20, 100);  
int hue = (int)random(80, 230); // int は「整数」を表す変数  
int saturation = 80;  
int brightness = 30;  
noStroke();  
fill(hue, saturation, brightness, 20);  
ellipse(x, y, w, h);  
//mouse button is pressed (マウスボタンがクリックされると)  
//stops Processing from continuously executing the code within draw()  
  (noLoop によってループが止まる)  
if (mousePressed == true) {  
  noLoop();  
}  
}  
  
//saves an image from the display window  
void keyPressed() {  
  if (key == ' ') {  
    save("P08.png");  
  }  
}
```

~~~~~  
練習 9 (P09) : 条件設定された繰り返し

[//https://processing.org/reference/](https://processing.org/reference/)

```
void setup() {  
  size(600, 600);  
  colorMode(HSB, 360, 100, 100, 100);  
  background(360);  
}
```

```

void draw() {
    //Blends the pixels in the display window according to a defined mode
    //BLEND, ADD, SUBTRACT, DARKEST, LIGHTEST, DIFFERENCE, EXCLUSION,
    MULTIPLY, SCREEN, REPLACE
    blendMode(DARKEST);
    //controls a sequence of repetitions (整数 i を使って繰り返し設定)
    for (int i = 0; i < 300; i++) {
        float x = random(width);
        float y = random(height);
        float w = random(20, 100);
        float h = random(20, 100);
        int hue = (int)random(0, 90);
        int saturation = 90;
        int brightness = 60;
        noStroke();
        fill(hue, saturation, brightness, 20);

        //modifies the location from which rectangles are drawn (四角形を描く基準点の位置)
        rectMode(CENTER);
        rect(x, y, w, h);
    }
    noLoop();
}

//saves an image from the display window
void keyPressed() {
    if ( key == ' ' ) {
        save( "P09.png" );
    }
}

```

~~~~~

練習 10 (P10) : 条件設定された繰り返しの中の繰り返し

[//https://processing.org/reference/](https://processing.org/reference/)

```

void setup() {
    size(600, 600);
    colorMode(HSB, 360, 100, 100, 100);

```



```

    background(360);
}
void draw() {
    //controls a sequence of repetitions (繰り返し処理の中に繰り返し処理)
    for (int i = 0; i <= width; i = i+50) {
        for (int j = 0; j <= height; j = j+50) {
            int hue = (int)random(0, 120);
            int saturation = 90;
            int brightness = 80;
            noStroke();
            fill(hue, saturation, brightness, 20);
            ellipse(i, j, 30, 30);
        }
    }
    noLoop();
}
//saves an image from the display window
void keyPressed() {
    if ( key == ' ' ) {
        save( "P10.png" );
    }
}

```

~~~~~

練習 1 1 (P11) : 条件設定された繰り返し (課題 3)

**//EXERCISE**

```

void setup() {
    size(800, 800);
    colorMode(HSB, 360, 100, 100, 100);
    background(360);
}

```

```

void draw() {

```

**//課題 3 : 以下を加筆修正して、繰り返し処理で描かれる図形を考える**

```

    for (int i = 3; i < 25; i++) {
        ellipse(20*i, 20*i, 20, 30);
    }
}

```

```

    }
    noLoop();
}
//saves an image from the display window
void keyPressed() {
    if ( key == ' ' ) {
        save( "Ex03.png" );
    }
}

```

~~~~~

練習 1 2 (P12) : 変数の活用

//変数を定義しておく

//変数を変えて結果の違いを確認する

```

int w = 80;
int h = 40;
int ellix = w/2;
int elliy = h/2;
int speedx = 1;
int speedy = 1;
int hue = 0;
void setup() {
    size(600, 600);
    colorMode(HSB, 360, 100, 100, 100);
    background(360);
}
void draw() {
    //background(360); (コメントアウトして塗りつぶさないようにしている)
    //変数 hue の値を 0~359 の 360 段階のループにする
    if (359 < hue) { //360 <= hue
        hue = 0;
    } else {
        hue = hue + 1;
    }
    //println(hue); (コンソールに書き出して変数の値を確認)
    int saturation = 99;

```

```

int brightness = 99;
noStroke();
fill(hue, saturation, brightness, 99);
ellipse(ellix, elliy, w, h);
//速度は変数の足し算で設定する
ellix+=speedx;//ellix=ellix+speedx;
elliy+=speedy;
//跳ね返りの設定では速度に-1 を掛ける
if (ellix + w/2 > width) {
    speedx = -speedx;
}
if (ellix - w/2 < 0) {
    speedx = -speedx;
}

if (elliy + h/2 > width) {
    speedy = -speedy;
}
if (elliy - h/2 < 0) {
    speedy = -speedy;
}
}
//saves an image from the display window
void keyPressed() {
    if ( key == ' ') {
        save( "P12.png" );
    }
}

```

---

練習 1 3 (p13) : オブジェクトの活用 (参考)

//オブジェクトを宣言

```

Ball b1, b2;
void setup() {
    size(600, 600);
    colorMode(HSB, 360, 100, 100, 100);

```

```

background(359);
//変数を渡してオブジェクトを生成
b1=new Ball(80, 40, 300, 300, 1, 1);
b2=new Ball(30, 60, 300, 300, 2, 1.5);
}

void draw() {
  //background(359);
  b1.update();
  b2.update();
}

//Ball クラスの設定
class Ball {
  int widthx, heighty, bx, by;
  int i =0;
  float speedx, speedy;

  //コンストラクタの設定
  Ball(int w, int h, int x, int y, float sx, float sy) {
    widthx = w;
    heighty = h;
    bx = x;
    by = y;
    speedx = sx;
    speedy = sy;
  }

  // draw()で毎回呼び出されて動く設定
  void update() {
    int hue = 360*i % 359;//割った余りでもループできる
    i+=1;
    //println(hue);
    int saturation = 99;
    int brightness = 99;
    noStroke();
    fill(hue, saturation, brightness, 99);
    ellipse(bx, by, widthx, heighty);
    bx += speedx;

```

```

    by += speedy;
    if (bx + widthx/2 > width) {
        speedx = -speedx;
    }
    if (bx - widthx/2 < 0) {
        speedx = -speedx;
    }
    if (by + heighty/2 > width) {
        speedy = -speedy;
    }
    if (by - heighty/2 < 0) {
        speedy = -speedy;
    }
}
}

//saves an image from the display window
void keyPressed() {
    if ( key == ' ') {
        save( "P13.png" );
    }
}

```

~~~~~

練習 1 4 (P14) : 配列の活用 (参考)

//オブジェクトの配列

Point[] tens = new Point[33];//個数を設定

```

void setup() {
    size(800, 800);
    colorMode(HSB, 360, 100, 100, 100);
    background(360);
    //オブジェクトの個数の分
    for (int i = 0; i < tens.length; i++) {
        int rw = int(random(10, 80));
        int rx = int(random(rw/2, width-rw/2));
        int ry = int(random(rw/2, height-rw/2));
        float rsx = random(1, 2);
    }
}

```

```

        float rsy = random(1.5, 3.5);
        tens[i] = new Point(rw, rx, ry, rsx, rsy);
    }
}

void draw() {
    //background(360);
    //オブジェクトの個数の分
    for (int i = 0; i < tens.length; i++) {
        tens[i].update();
    }
}

class Point {
    int sw, px, py;
    int i = int(random(360));
    float speedx, speedy;

    Point(int strokew, int bx, int by, float sx, float sy) {
        sw = strokew;
        px = bx;
        py = by;
        speedx = sx;
        speedy = sy;
    }

    void update() {
        int hue = 362*i % 361;
        i += 1;
        //println(hue);
        int saturation = 100;
        int brightness = 100;
        strokeWeight(sw);
        stroke(hue, saturation, brightness, 100);
        point(px, py);
        px += speedx;
        py += speedy;
        if (px > width) {
            speedx = -speedx;

```

```

    }
    if (px < 0) {
        speedx = -speedx;
    }

    if (py > width) {
        speedy = -speedy;
    }
    if (py < 0) {
        speedy = -speedy;
    }
}
}

//saves an image from the display window
void keyPressed() {
    if (key == ' ') {
        save("P13.png");
    }
}

```

~~~~~

練習 1 5 (P15) : 作品化 (参考)

```
En[] ens = new En[33];
```

```

void setup() {
    size(800, 800);
    colorMode(HSB, 360, 100, 100, 100);
    background(359);
    for (int i = 0; i < ens.length; i++) {
        int rw = int(random(10, 70));
        int rx = int(random(rw/2, width-rw/2));
        int ry = int(random(rw/2, height-rw/2));
        float rsx = random(1, 2);
        float rsy = random(1.5, 3.5);
        ens[i] = new En(rw, rx, ry, rsx, rsy);
    }
}

```

```
}
```

```
void draw() {  
    //background(359);  
    for (int i = 0; i < ens.length; i++) {  
        ens[i].update();  
    }  
}
```

```
class En {
```

```
    int sw, px, py;  
    int i = int(random(359));  
    float speedx, speedy;  
    float kuji = random(-1, 1);  
    int un = 1;
```

```
    En(int sizew, int bx, int by, float sx, float sy) {  
        sw = sizew;  
        px = bx;  
        py = by;  
        if (kuji < 0) {  
            un = -un;  
        } else {  
            un = 1;  
        }  
        speedx = sx*un;  
        speedy = sy*un;  
    }
```

```
    void update() {  
        int hue = 360*i % 359;  
        i+=1;  
        //println(hue);  
        int saturation = 99;  
        int brightness = 99;
```



```

    fill(hue, saturation, brightness, 99);
    ellipse(px, py, sw, sw);
    px += speedx;
    py += speedy;
    if (px>width) {
        speedx = -speedx;
    }
    if (px<0) {
        speedx = -speedx;
    }
    if (py>width) {
        speedy = -speedy;
    }
    if (py<0) {
        speedy = -speedy;
    }
}
}

//saves an image from the display window
void keyPressed() {
    if ( key == ' ' ) {
        save( "P15.png" );
    }
}

```