Design Programming

Spring 2018 - 8

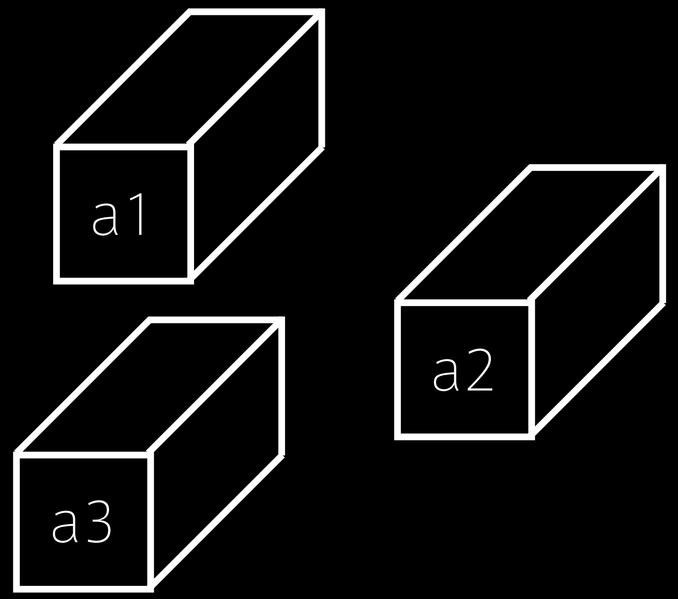
2018.6.6 Keio University, SFC

配列とは

同じ名前で格納されているデータの集まり

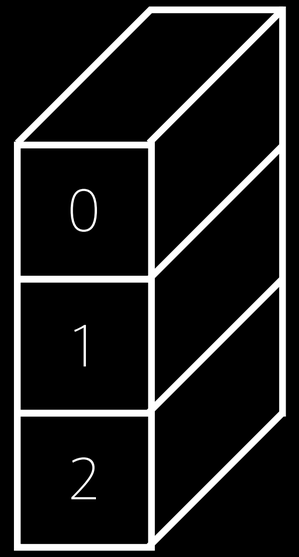
変数

a1, a2, a3



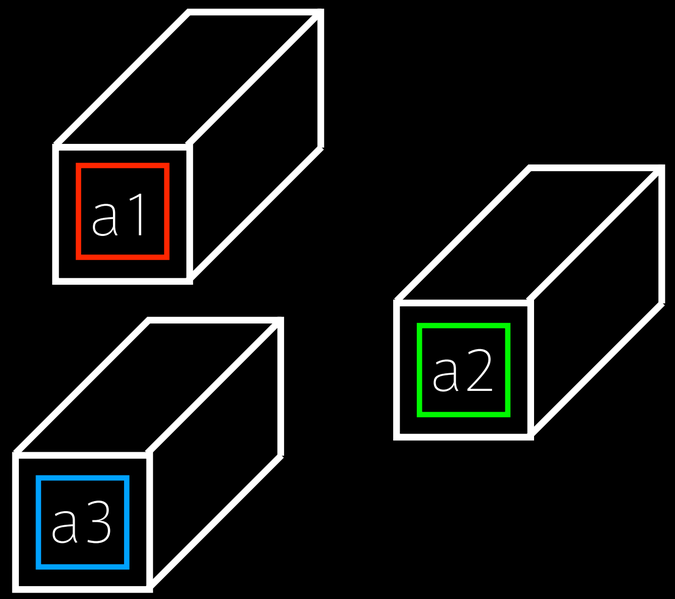
配列

a[3]



配列の使い方

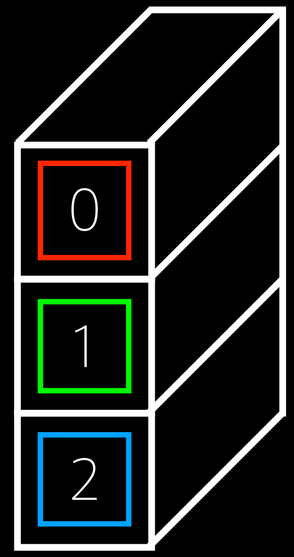
変数




```
int a1 = 0, a2 = 0, a3 = 0;
a1 = 10;
a2 = 20;
a3 = 30;
background(a1);
```

08/variable/variable.pde

配列

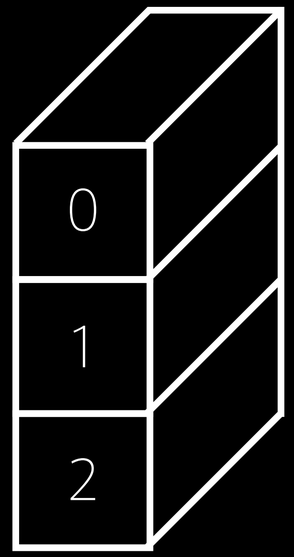



```
int[] a;
a = new int[3];
a[0] = 10;
a[1] = 20;
a[2] = 30;
background(a[0]);
```

要素数を指定して宣言

08/array/array.pde

配列の使い方




```
int[] a = {10, 20, 30};
background(a[0]);
```

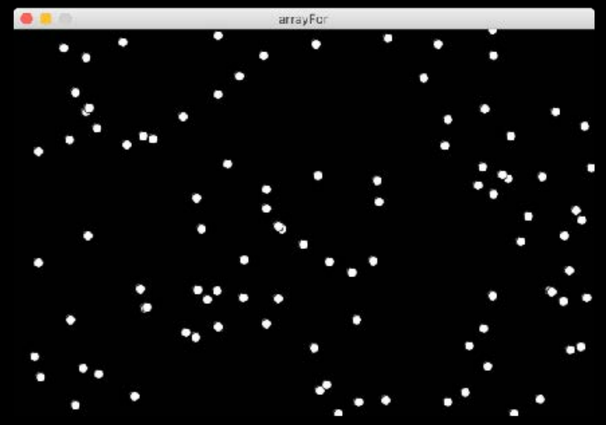
一度に直接要素を指定することも可能

08/array2/array2.pde

for文と組み合わせる

```
int[] x = new int[100];
int[] y = new int[100];

void setup() {
    size(600, 400);
    for (int i = 0; i < x.length; i++) {
        x[i] = (int)random(width);
        y[i] = (int)random(height);
    }
}

void draw() {
    background(0);
    for (int i = 0; i < x.length; i++) {
        ellipse(x[i], y[i], 10, 10);
    }
}
```

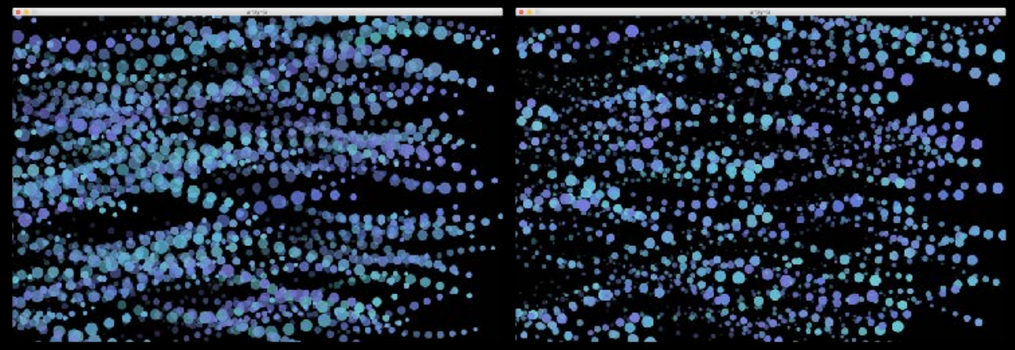
配列の要素数（x.length）の条件でfor文をまわす



08/arrayFor/arrayFor.pde

for文と自作関数と組み合わせる

```
int el = 200;
int[] n = new int[el];
int[] x = new int[el];
int[] y = new int[el];
float[] sp = new float[el];
color[] c = new color[el];

void setup() {
    size(1200, 800);
    for (int i = 0; i < x.length; i++) {
        n[i] = (int)random(10, 20);
        x[i] = (int)random(width);
        y[i] = (int)random(height);
        sp[i] = random(4.0);
        c[i] = color(random(100, 120), random(120, 200), random(200, 220));
    }
}

void draw() {
    background(0);
    for (int i = 0; i < x.length; i++) {
        nyoro(n[i], x[i], y[i], c[i], sp[i], true);
    }
}
```



08/arrayForFunction/arrayForFunction.pde

関数以下略

独立したアニメーション

初期位置x
初期位置y
横方向初期速度vx
横方向初期速度vy
円の径d
円の色c

のインデックスi

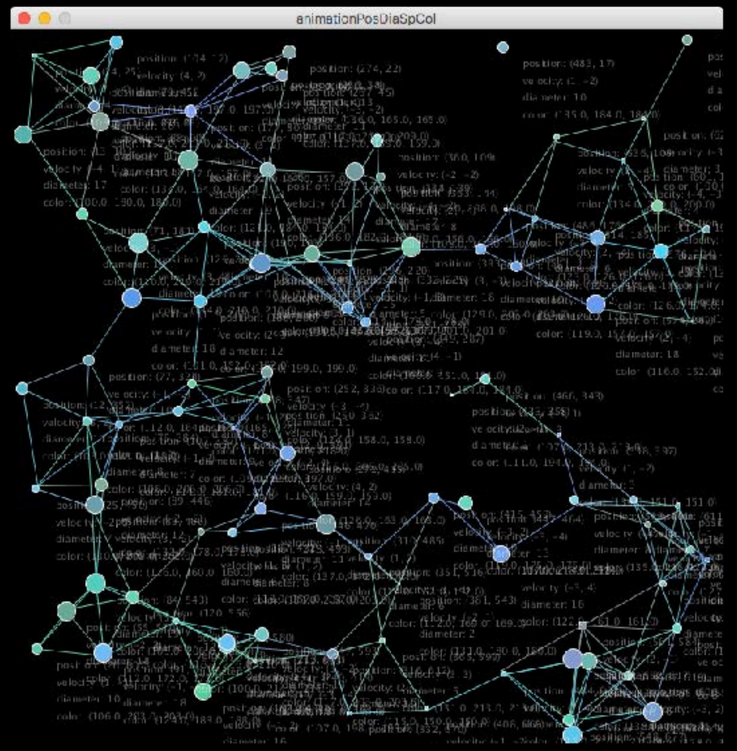
円が画面内で跳ね返る

のインデックスi

情報をテキストで書く

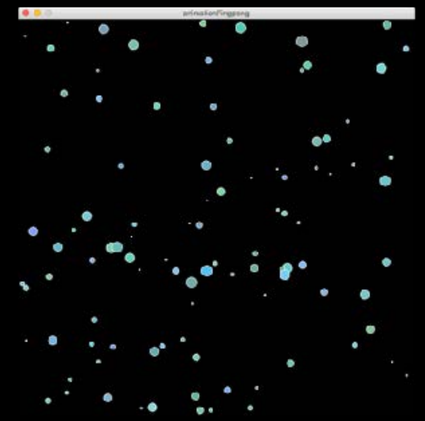
のインデックスi

近いポイントを線でつなぐ



独立したアニメーション

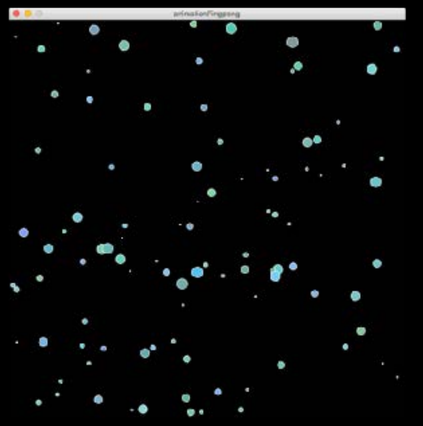
```
int n = 10;
int[] x = new int[n];
int[] y = new int[n];
int[] d = new int[n];
int[] vx = new int[n];
int[] vy = new int[n];
color[] c = new color[n];

void setup() {
    size(700, 700);
    for (int i = 0; i < x.length; i++) {
        d[i] = (int)random(2, 20);
        x[i] = (int)random(d[i] / 2, width - d[i] / 2);
        y[i] = (int)random(d[i] / 2, height - d[i] / 2);
        while (vx[i] == 0) vx[i] = (int)random(-5, 5);
        while (vy[i] == 0) vy[i] = (int)random(-5, 5);
        c[i] = color(random(100, 140), random(150, 220), random(150, 250));
    }
}
```



08/animationPingpong/animationPingpong.pde

独立したアニメーション

```
void draw() {
    background(0);
    for (int i = 0; i < x.length; i++) {
        pingpong(i);
    }
}

void pingpong(int i) {
    if (x[i] >= width - d[i] / 2 || x[i] <= d[i] / 2)
        vx[i] = -vx[i];
    if (y[i] >= height - d[i] / 2 || y[i] <= d[i] / 2)
        vy[i] = -vy[i];
    x[i] += vx[i];
    y[i] +=vy[i];

    fill(c[i]);
    stroke(255);
    ellipse(x[i], y[i], d[i], d[i]);
}
```



08/animationPingpong/animationPingpong.pde

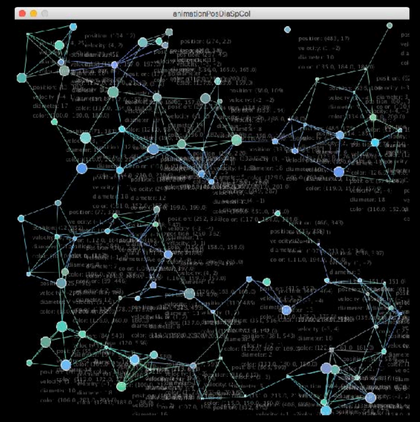
独立したアニメーション

```
void draw() {
    textSize(10);
    background(0);
    for (int i = 0; i < x.length; i++) {
        pingpong(i);
        info(i);
    }
}

void info(int i) {
    String s;
    s = "position: (" + x[i] + ", " + y[i] + ")";
    s += '\n' + "velocity: (" + vx[i] + ", " + vy[i] + ")";
    s += '\n' + "diameter: " + d[i];
    s += '\n' + "color: (" + red(c[i]) + ", " + green(c[i]) + ", " +
green(c[i]) + ")";
    fill(255, 100);
    text(s, x[i] + 20, y[i] + 20);
}
```

08/animationInfo/animationInfo.pde

独立したアニメーション

```
void draw() {
    background(0);
    for (int i = 0; i < x.length; i++) {
        pingpong(i);
        connect(i);
        info(i);
    }
}

void connect(int i) {
    stroke(c[i]);
    for (int j = 0; j < n; j++) {
        if (i != j && i < j) {
            if (dist(x[i], y[i], x[j], y[j]) <= 100)
                line(x[i], y[i], x[j], y[j]);
        }
    }
}
```



08/animationConnect/animationConnect.pde

演習 1

これまでのスケッチのpingpong関数の代わりに,

マウスとの距離が200pixel以内のときに引き寄せられる関数

mouseAttractionForceを書き，描画しなさい.

ヒント

```
void mouseAttractionForce(int i) {
    if (dist(                                    ) <=      ) {
        vx[i] +=                                  ? 1 : -1;
        vy[i] +=                                  ? 1 : -1;
        x[i] += vx[i];
        y[i] += vy[i];
    }
    fill(c[i]);
    stroke(255);
    ellipse(                                        );
}
```

ちなみに. .

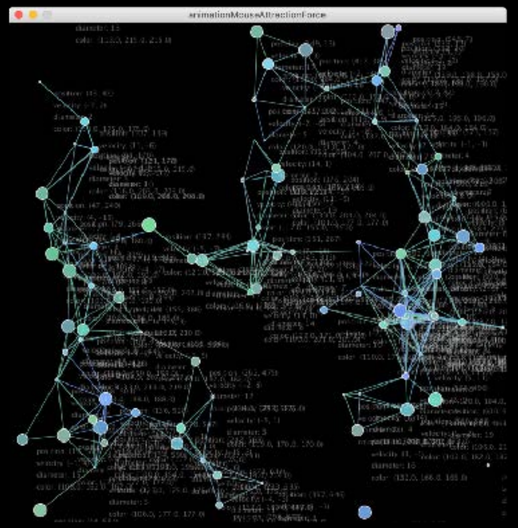
```
A ? B : C;
```

Aという条件が

真ならばB，偽ならばCを返す

演習 1

```
void mouseAttractionForce(int i) {
    if (dist(mouseX, mouseY, x[i], y[i]) <= 200) {
        vx[i] += (mouseX - x[i]) >= 0 ? 1 : -1;
        vy[i] += (mouseY - y[i]) >= 0 ? 1 : -1;
        x[i] += vx[i];
        y[i] += vy[i];
    }
    fill(c[i]);
    stroke(255);
    ellipse(x[i], y[i], d[i], d[i]);
}
```



08/animationmouseAttractionForce/animationmouseAttractionForce.pde

課題

- 今日学習した「配列とアニメーション」を使用してスケッチを１つアップロードしなさい.
  アップロード先は「08 array」とする.
  ※今日学習していない分野でもすでに自分が知っている技術は使用可能とする.

  〆切：6月12日24時まで

次回

- オブジェクト指向プログラミング