

モバイルシステム演習

Processing

- 関数 -

関数の使い所

- ・ 同じような処理を繰り返す場合
- ・ 一つの機能(モジュール)として処理できる場合



関数でまとめる

関数の定義①

引数を伴わない場合の関数定義

返り値なしの型 関数名 引数なし

```
void fadeToWhite() {  
  noStroke();  
  fill(255, 30);  
  rectMode(CORNER);  
  rect(0, 0, width, height);  
}
```

関数の定義①

引数を伴う場合の関数定義

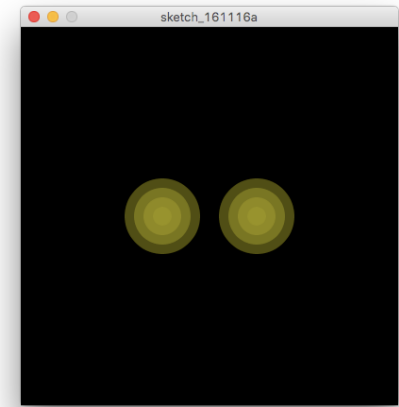
戻り値の型	関数名	引数
float	add	(float x, float y)

```
{  
    return x + y;  
}
```

使用例

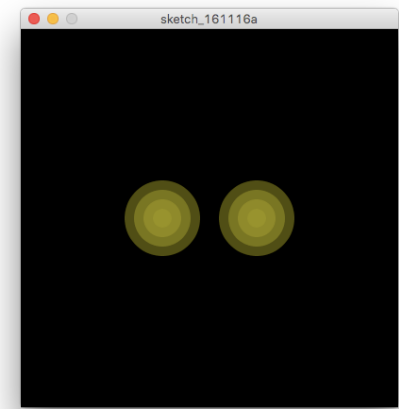


このように記述できる



```
void draw() {  
  int centerX = 150;  
  int centerY = 200;  
  for (int i = 0; i < 5; i++) {  
    fill(255, 255, 0, i * 20);  
    ellipse(centerX, centerY, i * 20, i * 20);  
  }  
  
  centerX = 250;  
  centerY = 200;  
  for (int i = 0; i < 5; i++) {  
    fill(255, 255, 0, i * 20);  
    ellipse(centerX, centerY, i * 20, i * 20);  
  }  
}
```

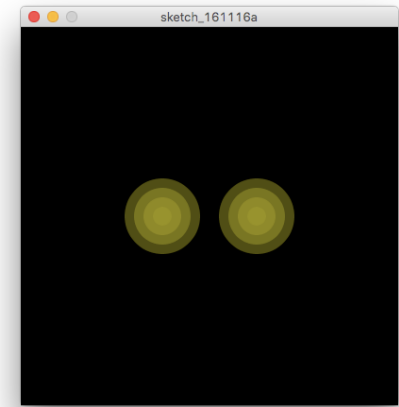
同じような処理をしている



```
void draw() {  
  int centerX = 150;  
  int centerY = 200;  
  for (int i = 0; i < 5; i++) {  
    fill(255, 255, 0, i * 20);  
    ellipse(centerX, centerY, i * 20, i * 20);  
  }  
}
```

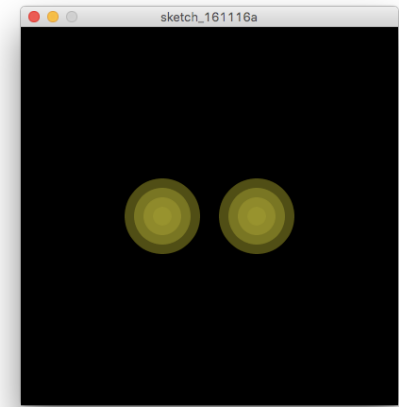
```
  centerX = 250;  
  centerY = 200;  
  for (int i = 0; i < 5; i++) {  
    fill(255, 255, 0, i * 20);  
    ellipse(centerX, centerY, i * 20, i * 20);  
  }  
}
```

関数を使ってまとめる①



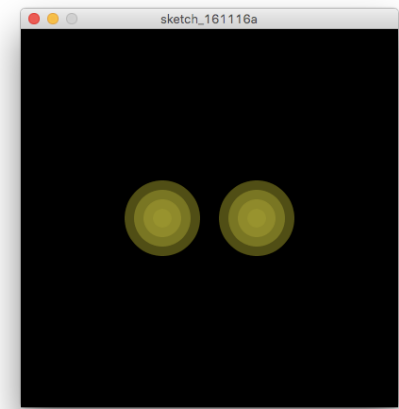
```
void drawCircles(int x, int y) {  
  for (int i = 0; i < 5; i++) {  
    fill(255, 255, 0, i * 20);  
    ellipse(x, y, i * 20, i * 20);  
  }  
}
```


関数を使ってまとめる②



```
void drawCircles(int x, int y) {  
  pushMatrix();  
  // 座標の原点を移動  
  translate(x, y);  
  for (int i = 0; i < 5; i++) {  
    fill(255, 255, 0, i * 20);  
    ellipse(0, 0, i * 20, i * 20);  
  }  
  popMatrix();  
}
```

関数を呼び出す



```
void draw() {  
  drawCircles(150, 200);  
  drawCircles(250, 200);  
}
```

同じ形の物体を関数を使って 大量に作成しよう

