

プロセツシング 入門1

初歩のプログラミング

プロセッシングに慣れることが1回目の目標
processing基礎最速入門, processing学習ノート
を参考にさせていただきました。

問1

ellipse(width/2, height/2, mouseX, mouseY);
を
rectMode(CENTER);
rect(width/2, height/2, mouseX, mouseY);
の2行に変更してみよ

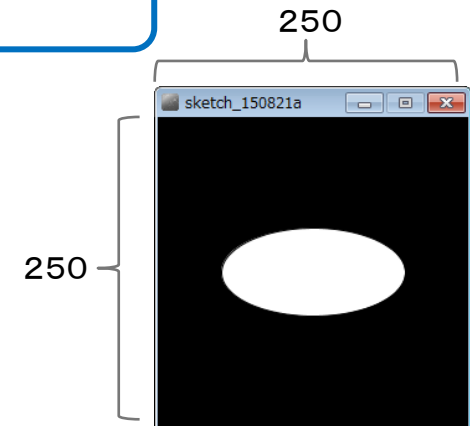
```
sketch_150821a | Processing 2.2.1
File Edit Sketch Tools Help
sketch_150821a
void setup() {
  size(250, 250);
  frameRate(20);
}
void draw() {
  background(0);
  ellipse(width/2, height/2, mouseX, mouseY);
}
```

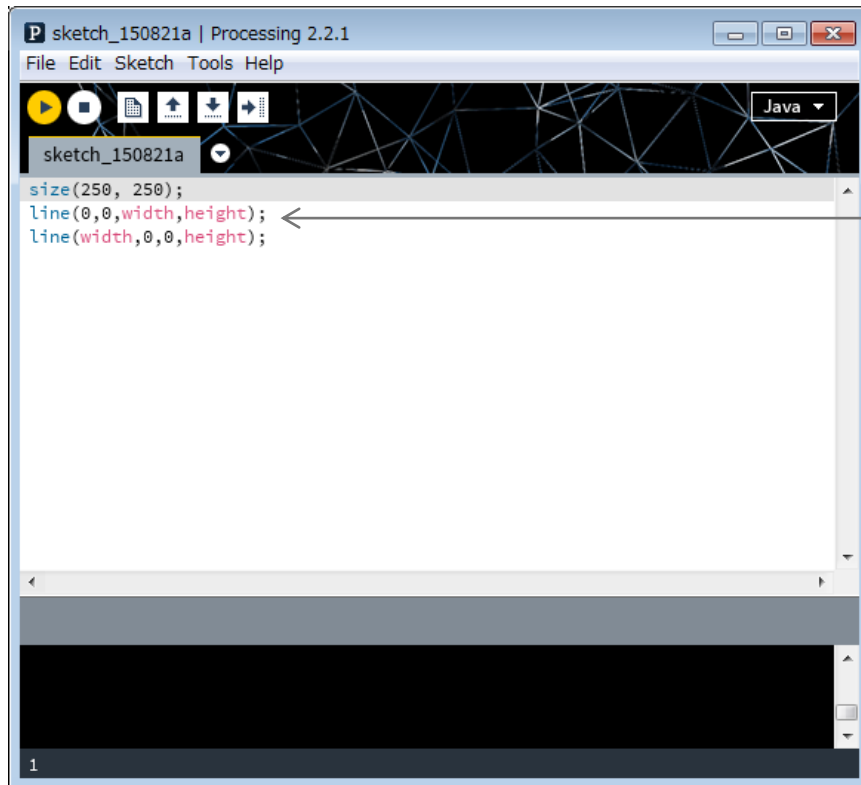
250×250のウィンドウサイズにするという意味

1秒間に20コマ描きます

width=250,height=250,mouseX(マウスのx座標), mouseY(マウスのy座標)です
中心の座標(width/2,height/2)でx方向の長さmouseX, y方向の長さmouseYの
楕円を描く

背景を黒にするという意味. 値を変えてみよう.

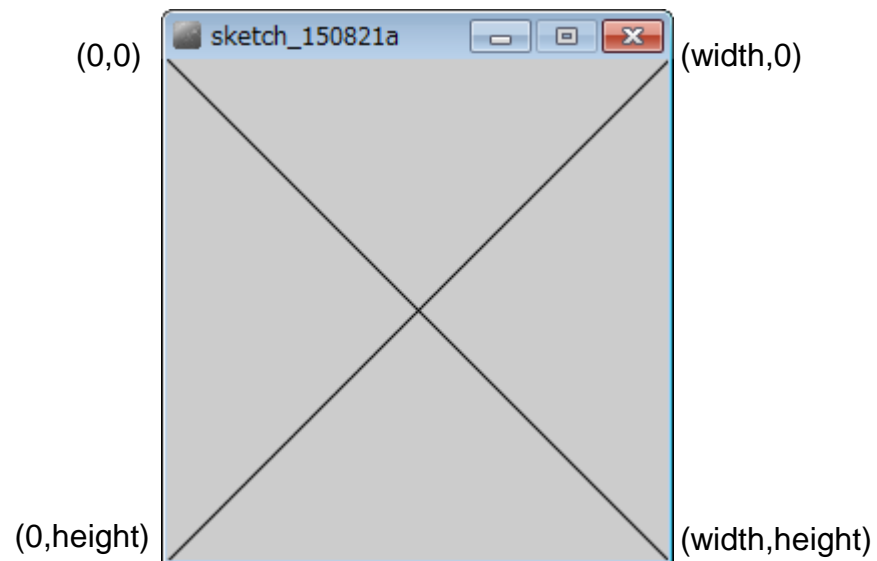




setup()とdraw()はアニメーションに必要
アニメーションの必要がなければ、使いません。

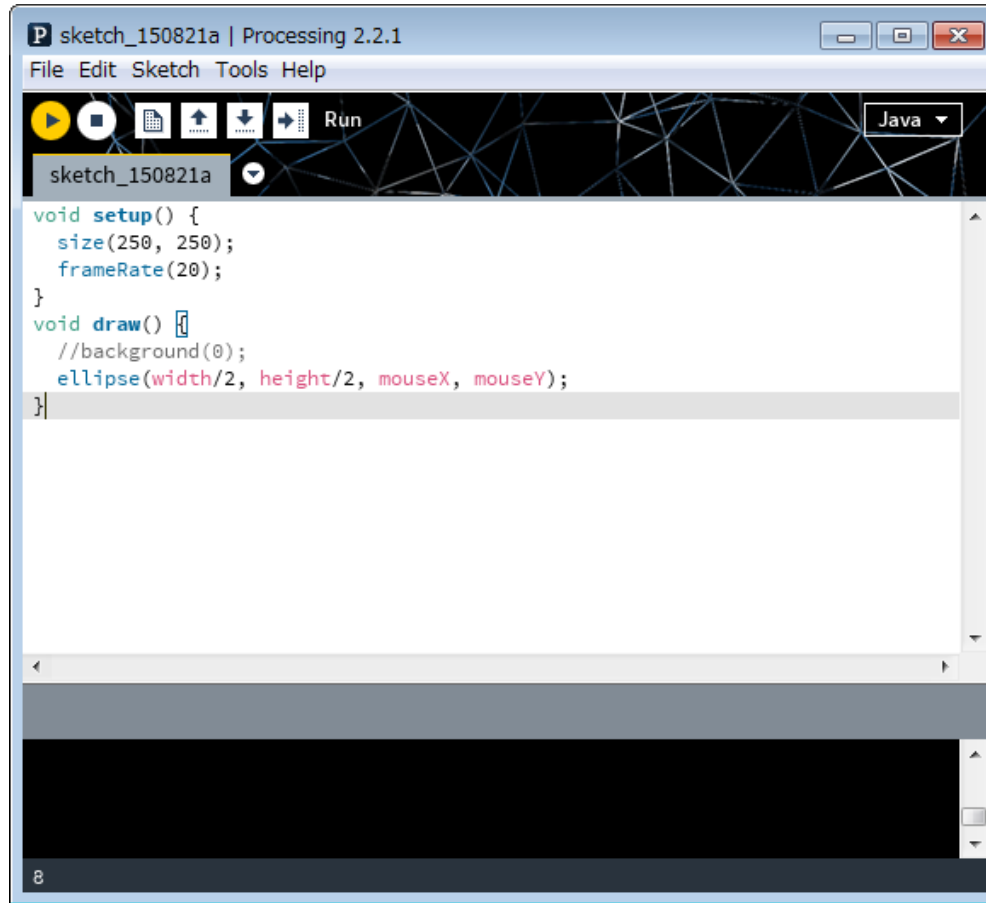
width=250, height=250です。

line(x, y, a, b)は点(x, y)から点(a, b)へ直線を引く
命令です。

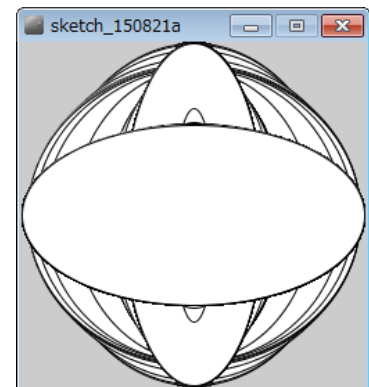


重ね塗り

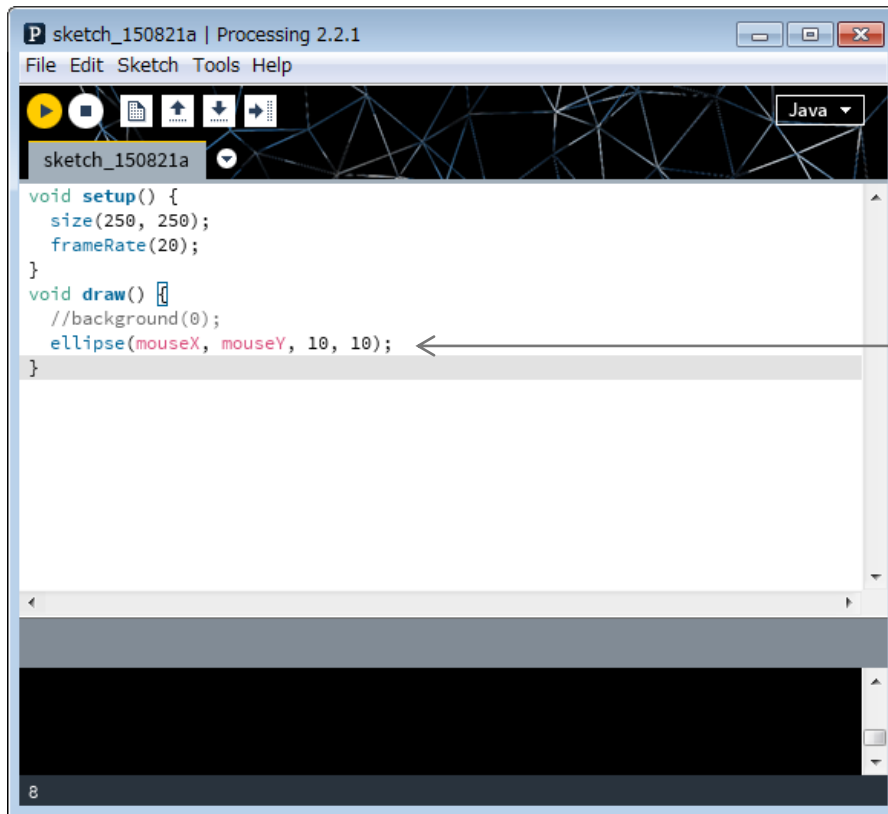
楕円を書いたプログラムでbackground(0)をコメントアウト(//でコマンドとして無効化すること)してみる。すると、毎回黒に塗りつぶすことを行わないことになる。どうなるだろうか？



```
sketch_150821a | Processing 2.2.1
File Edit Sketch Tools Help
Run
sketch_150821a
void setup() {
  size(250, 250);
  frameRate(20);
}
void draw() {
  //background(0);
  ellipse(width/2, height/2, mouseX, mouseY);
}
```



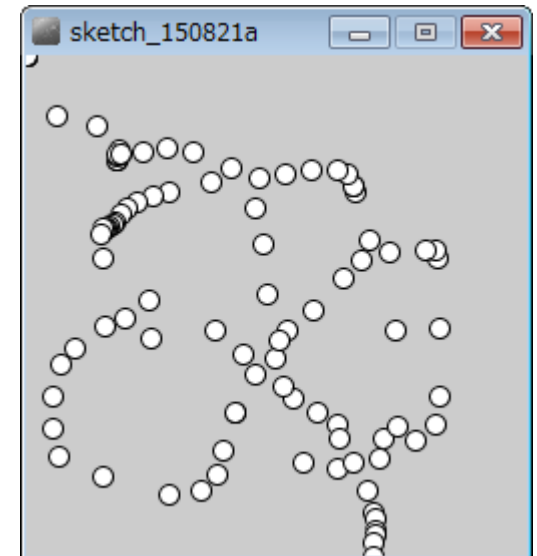
マウスの位置に描く



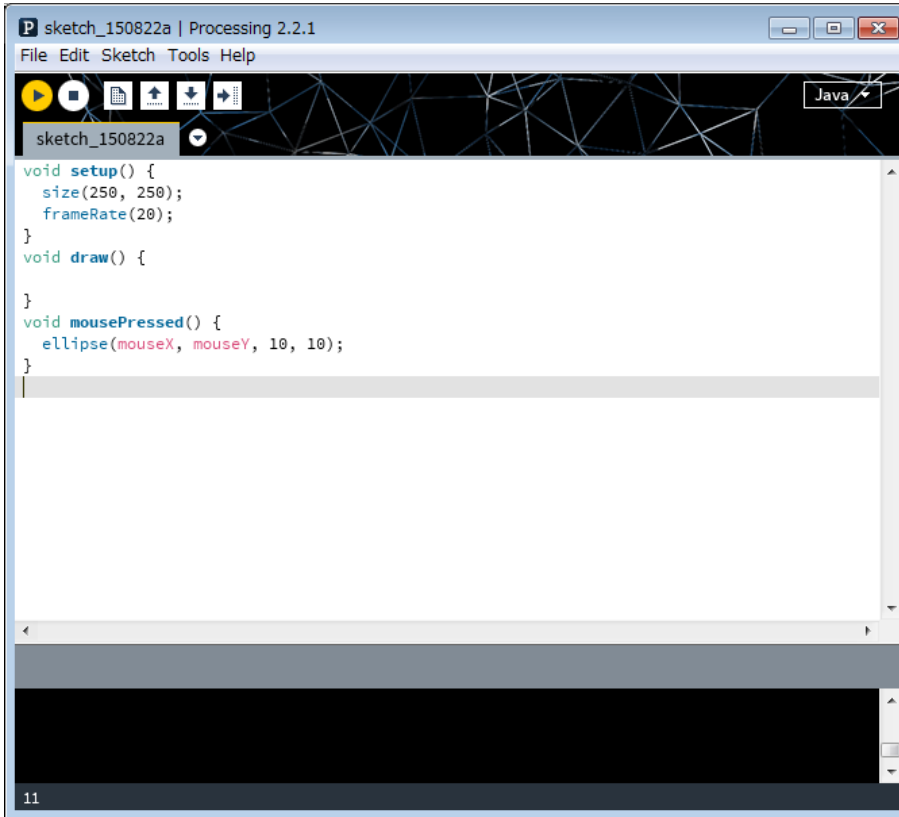
```
void setup() {  
  size(250, 250);  
  frameRate(20);  
}  
void draw() {  
  //background(0);  
  ellipse(mouseX, mouseY, 10, 10);  
}
```

マウスの(x, y)座標(mouseX, mouseY)の位置に半径10の円を描くことになります。

問2
半径10の円を一辺10の正方形に変更してみよ。

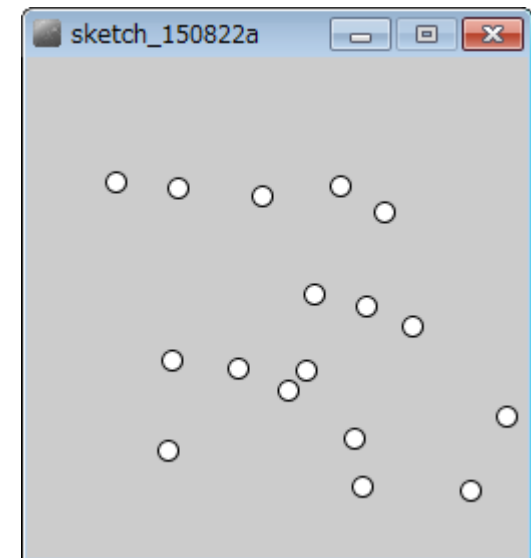


クリックしたときだけ描く(1)



```
void setup() {  
  size(250, 250);  
  frameRate(20);  
}  
void draw() {  
  
}  
void mousePressed() {  
  ellipse(mouseX, mouseY, 10, 10);  
}
```

mousePressed()は、マウスのクリック時の動作を設定する命令です。draw()以下の{...}は、フレームレートごとに呼び出されていましたが、mousePressed()以下の{...}は、マウスの左ボタンを押したときだけ実行されます。これを**割り込み処理**といいます。今は気にしないでよいです。



クリックしたときだけ描く(2)

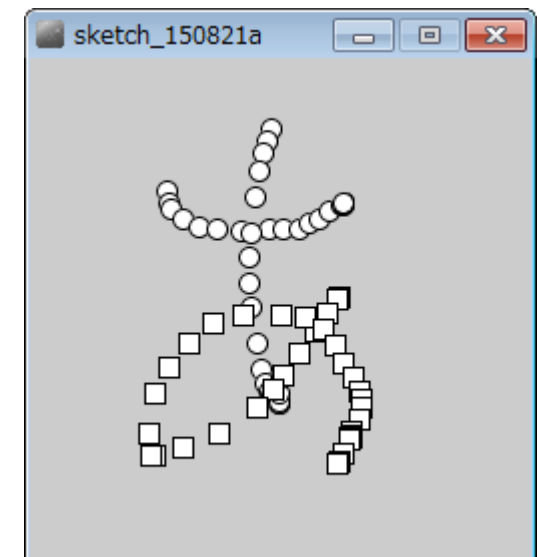
```
sketch_150821a | Processing 2.2.1
File Edit Sketch Tools Help
sketch_150821a
void setup() {
  size(250, 250);
  frameRate(20);
}
void draw() {
  if(mousePressed&&(mouseButton == LEFT)){
    ellipse(mouseX, mouseY, 10, 10);
  }
  if(mousePressed&&(mouseButton == RIGHT)){
    rect(mouseX, mouseY, 10, 10);
  }
}
```

今度は、draw()中でマウスボタンが押されたかどうかの判定を行ってみます。

「マウスのボタンが押されたら」という意味です。

&&は「かつ」を表します。ですから「マウスボタンが押されてかつそれが左ボタンだったら」を表します。

左ボタンを押している間は、○で描画され、右ボタンを押している間は□で描画されることを確かめましょう。



問3

マウスの右ボタンを押すと描画がクリアされるように変更してみよ。
ヒント: setup()と右ボタンを押された時の処理にbackground(125)を加える。コピペ帳に答えはありますが、まず考えてみよう。