int stepX;

int stepY;

🡨 Variable definition: type name

**void setup**(){

size(800, 400); 🡨 The display size

background(0);

}

**void draw**(){

stroke(13);

🡨 Sets the color used to draw lines and borders around shapes. This color is either specified in terms of the RGB or HSB color depending on the current colorMode() (the default color space is RGB, with each value in the range from 0 to 255).

stepX = mouseX+1;

stepY = mouseY+1;

🡨 The system variable mouseX always contains the current horizontal coordinate of the mouse, and the mouseY contains the current vertical coordinate of the mouse. The addition of 2 prevent stepX or stepY from being too small, which lead to longer display times

for (int gridY = 0; gridY < height; gridY+=stepY){

rect(0, gridY, stepX, stepY);

}

void draw(){

stroke(13);

stepX = mouseX+1;

stepY = mouseY+1;

for (int gridY=0; gridY < height; gridY+=stepY){

rect(0, gridY, stepX, stepY);

}

**for (int gridX=0; gridX < width; gridX+=stepX){**

**rect(gridX, 0, stepX, stepY);**

**}**

}

void draw(){

stroke(13);

stepX = mouseX+1;

stepY = mouseY+1;

for (int gridY = 0; gridY < height; gridY+=stepY){

**for (int gridX = 0; gridX < width; gridX+=stepX){**

**rect(gridX, gridY, stepX, stepY);**

**}**

}

}

void draw(){

stroke(13);

colorMode(HSB, width, height, 100);

stepX = mouseX+1;

stepY = mouseY+1;

for (int gridY = 0; gridY < height; gridY+=stepY){

for (int gridX = 0; gridX < width; gridX+=stepX){

**fill(gridX, height-gridY, 100);**

rect(gridX, gridY, stepX, stepY);

}

}

}

void draw(){

**noStroke();**

**colorMode(HSB, width, height, 100);**

stepX = mouseX+1;

stepY = mouseY+1;

for (int gridY = 0; gridY < height; gridY+=stepY){

for (int gridX = 0; gridX < width; gridX+=stepX){

**fill(gridX, height-gridY, 100);**

rect(gridX, gridY, stepX, stepY);

}

}

}