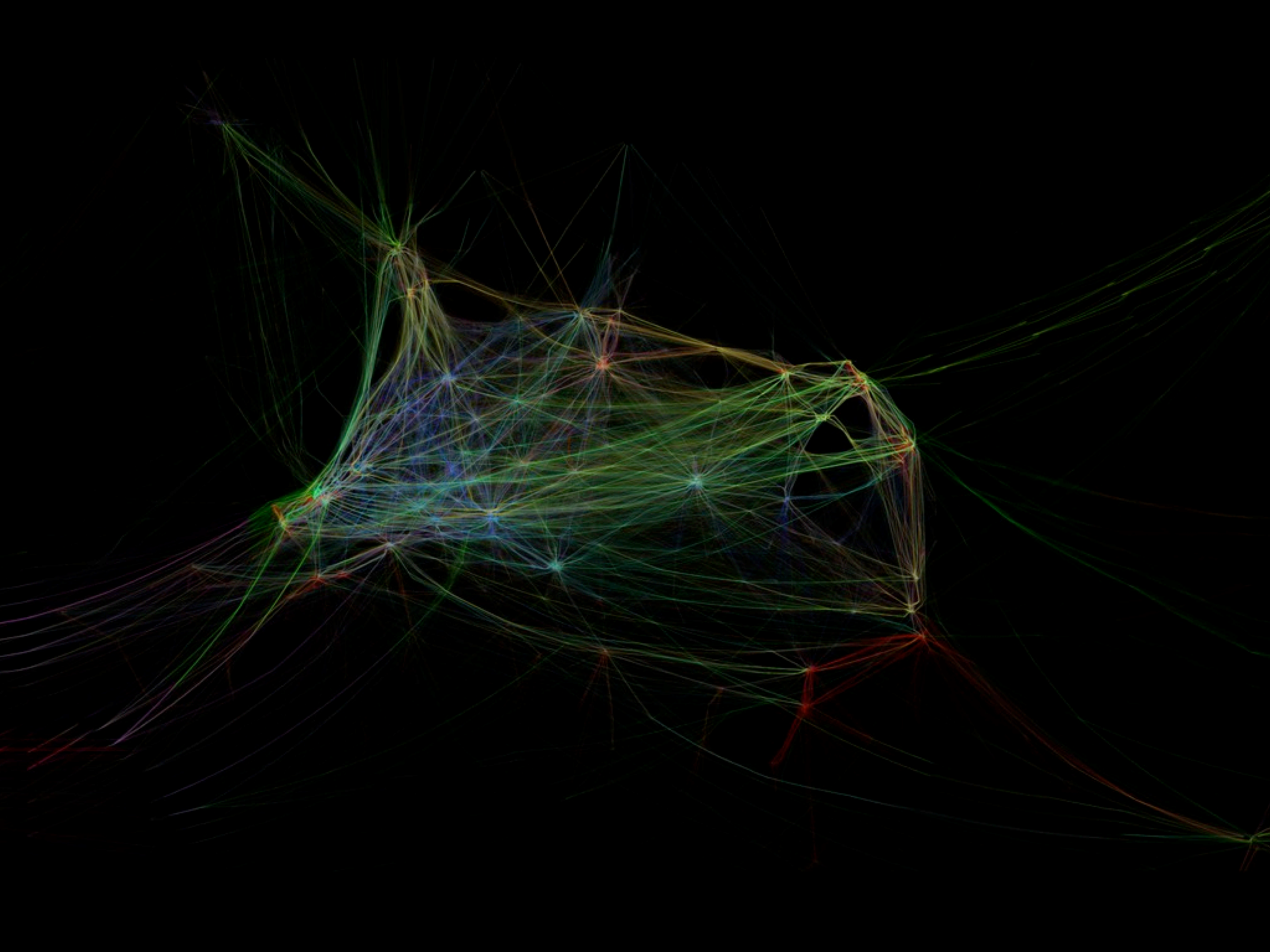
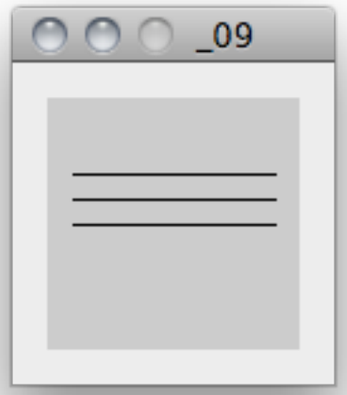
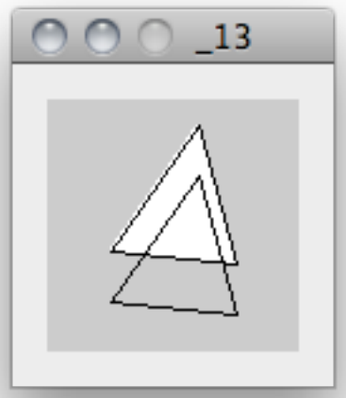
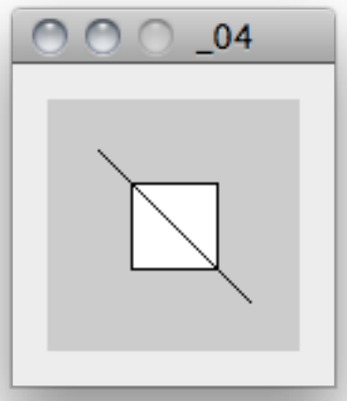
PROCESSING




_09


_13


_04

Computer sind dumm

Computer sind pedantisch

```
size(200, 200); // Runs the size() function
int x; // Declares a new variable x
x = 102; // Assigns the value 102 to the variable x
background(x); // Runs the background() function
```

```
size(200, 200); // Runs the size() function
int x; // Declares a new variable x
x = 102; // Assigns the value 102 to the variable x
background(x); // Runs the background() function
```

```
size(200, 200); // Runs the size() function
int x; // Declares a new variable x
x = 102; // Assigns the value 102 to the variable x
background(x); // Runs the background() function
```

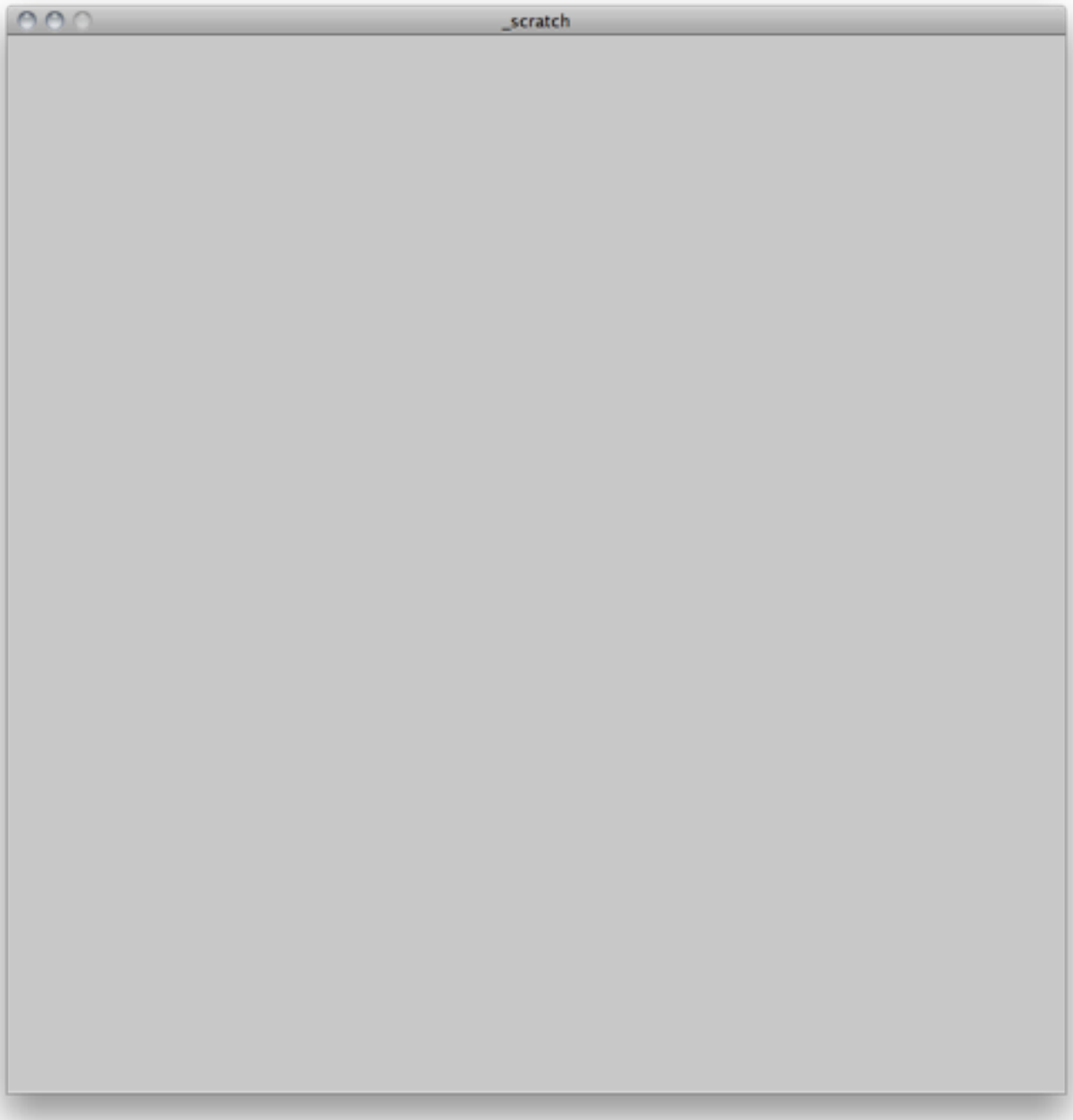
```
size(200, 200); // Runs the size() function
int x; // Declares a new variable x
x = 102; // Assigns the value 102 to the variable x
background(x); // Runs the background() function
```

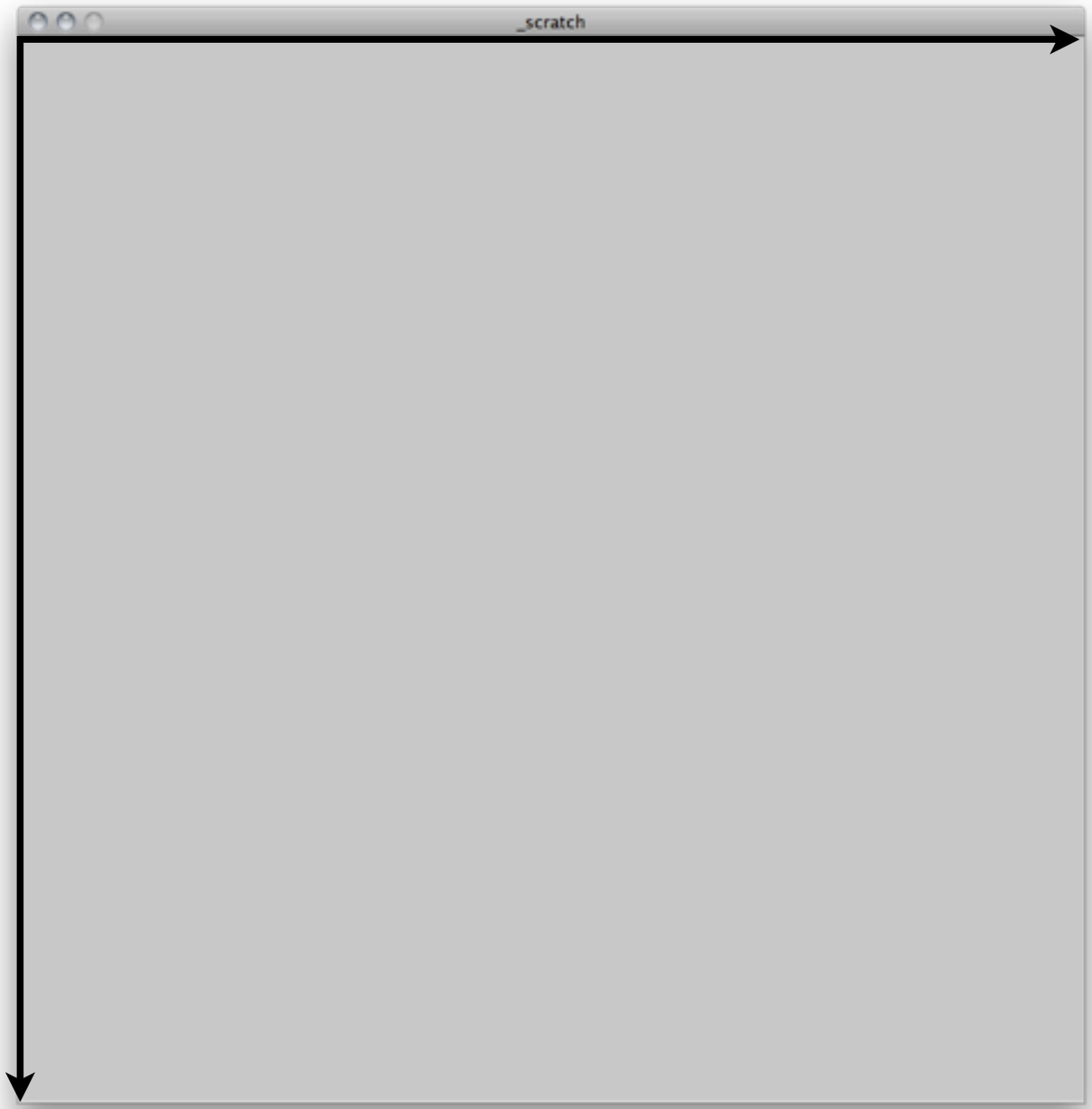
```
size(200, 200);
Background(102);
// ERROR! The B in "background" is capitalized
```

```
// To print text to the screen, place the desired output in quotes
println("Processing..."); // Prints "Processing..." to the console
```

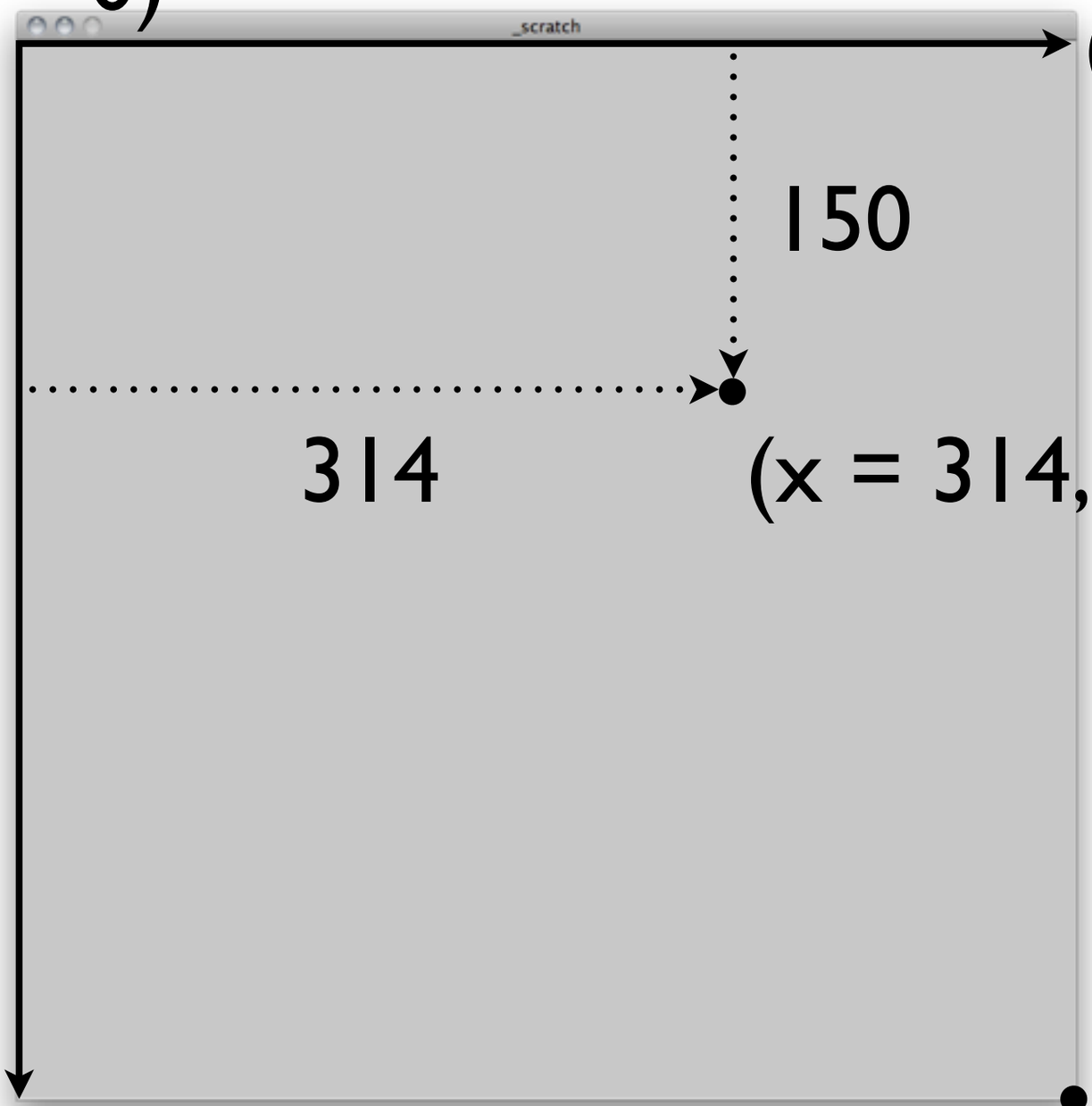
```
// To print the value of a variable, rather than its name, don't put
// the name of the variable in quotes.
int x = 20;
println(x); // Prints "20" to the console
println("x"); // Prints "x" to the console
```

```
// The "+" operator can be used for combining multiple text
// elements into one line.
int x2 = 20;
int y2 = 80;
println(x2 + " : " + y2);
// Prints "20 : 80" to the message window
```


_scratch


_scratch

(x = 0, y = 0)



(x = 470, y = 0)

(x = 0, y = 470)

(x = 470, y = 470)

```
// Left line
line(10, 80, 30, 40);

line(20, 80, 40, 40);

// Middle line
line(30, 80, 50, 40);

line(40, 80, 60, 40);

// Right line
line(50, 80, 70, 40);
```

```
// Left line
line(10, 80, 30, 40);

line(20, 80, 40, 40);

// Middle line
line(30, 80, 50, 40);

line(40, 80, 60, 40);

// Right line
line(50, 80, 70, 40);
```

```
// Left line
line(10, 80, 30, 40);

line(20, 80, 40, 40);

// Middle line
line(30, 80, 50, 40);

line(40, 80, 60, 40);

// Right line
line(50, 80, 70, 40);
```

```
// Left line
line(10, 80, 30, 40);

line(20, 80, 40, 40);

// Middle line
line(30, 80, 50, 40);

line(40, 80, 60, 40);

// Right line
line(50, 80, 70, 40);
```

```
// Left line
line(10, 80, 30, 40);

line(20, 80, 40, 40);

// Middle line
line(30, 80, 50, 40);

line(40, 80, 60, 40);

// Right line
line(50, 80, 70, 40);
```

```
// Left line
line(10, 80, 30, 40);

line(20, 80, 40, 40);

// Middle line
line(30, 80, 50, 40);

line(40, 80, 60, 40);

// Right line
line(50, 80, 70, 40);
```

```
// Left line
line(10, 80, 30, 40);

line(20, 80, 40, 40);

// Middle line
line(30, 80, 50, 40);

line(40, 80, 60, 40);

// Right line
line(50, 80, 70, 40);
```

```
background(0); // Sets the black background
stroke(255); // Sets line value to white
strokeWeight(5); // Sets line width to 5 pixels
// Makes the lines draw with smooth edges
smooth();

line(10, 80, 30, 40); // Left line
line(20, 80, 40, 40);
line(30, 80, 50, 40); // Middle line
line(40, 80, 60, 40);
line(50, 80, 70, 40); // Right line
```

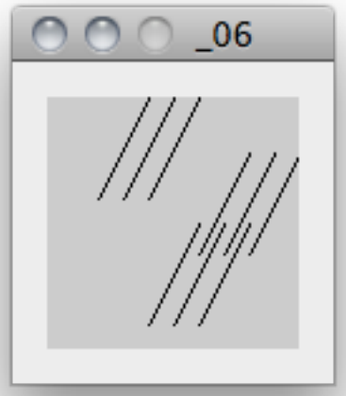
```
int x = 5; // Sets the horizontal position of the lines
int y = 60; // Sets the vertical position of the lines

// Draws line from [5,60] to [25,20]
line(x, y, x+20, y-40);
// Draws line from [15,60] to [35,20]
line(x+10, y, x+30, y-40);
// Draws line from [25,60] to [45,20]
line(x+20, y, x+40, y-40);
// Draws line from [35,60] to [55,20]
line(x+30, y, x+50, y-40);
// Draws line from [45,60] to [65,20]
line(x+40, y, x+60, y-40);
```

```
// Sets the horizontal position of the lines
int x;
x = 0;
```

```
// Sets the horizontal position of the lines
int x;
x = 0;
```

```
// Sets the horizontal position of the lines
int x;
x = 0;
```

```
// Sets the horizontal position of the lines
int x;
x = 0;
// Sets the vertical position of the lines
int y = 55;
```


_06

```
void diagonals(int x, int y)
{
   line(x, y, x+20, y-40);
   line(x+10, y, x+30, y-40);
   line(x+20, y, x+40, y-40);
}
```

```
void diagonals(int x, int y)
{
   line(x, y, x+20, y-40);
   line(x+10, y, x+30, y-40);
   line(x+20, y, x+40, y-40);
}
```

```
void diagonals(int x, int y)
{
   line(x, y, x+20, y-40);
   line(x+10, y, x+30, y-40);
   line(x+20, y, x+40, y-40);
}
```

```
void diagonals(int x, int y)
{
   line(x, y, x+20, y-40);
   line(x+10, y, x+30, y-40);
   line(x+20, y, x+40, y-40);
}
```

```
void diagonals(int x, int y)
{
   line(x, y, x+20, y-40);
   line(x+10, y, x+30, y-40);
   line(x+20, y, x+40, y-40);
}
```

```
void diagonals(int x, int y)
{
   line(x, y, x+20, y-40);
   line(x+10, y, x+30, y-40);
   line(x+20, y, x+40, y-40);
}
```

```
void diagonals(int x, int y)
{
   line(x, y, x+20, y-40);
   line(x+10, y, x+30, y-40);
   line(x+20, y, x+40, y-40);
}
```

```
void diagonals(int x, int y)
{
    line(x, y, x+20, y-40);
    line(x+10, y, x+30, y-40);
    line(x+20, y, x+40, y-40);
}
```

```
void setup() {
  size(100, 100);
  noLoop();
}

void draw() {
  diagonals(40, 90);
  diagonals(60, 62);
  diagonals(20, 40);
}

void diagonals(int x, int y) {
  line(x, y, x+20, y-40);
  line(x+10, y, x+30, y-40);
  line(x+20, y, x+40, y-40);
}
```

```
void setup() {
  size(100, 100);
  noLoop();
}

void draw() {
  diagonals(40, 90);
  diagonals(60, 62);
  diagonals(20, 40);
}

void diagonals(int x, int y) {
  line(x, y, x+20, y-40);
  line(x+10, y, x+30, y-40);
  line(x+20, y, x+40, y-40);
}
```

```
void setup() {
  size(100, 100);
  noLoop();
}

void draw() {
  diagonals(40, 90);
  diagonals(60, 62);
  diagonals(20, 40);
}

void diagonals(int x, int y) {
  line(x, y, x+20, y-40);
  line(x+10, y, x+30, y-40);
  line(x+20, y, x+40, y-40);
}
```

http://j.mp/7Um73T