Processing Workshop

Friday Night Edition

Nathan Lachenmyer

CEMI Electronic Media Institute

September 28, 2012

What is Processing?

Processing is a programming language specifically designed to **generate** and **modify** images.

What is Processing?

Processing is a software sketchbook – it makes it easy to **explore** and **refine** ideas **quickly**.

What is Processing?

Processing was designed to engage people with **visual** and **spatial** minds, to open up programming to **artists** and **designers**.

About Me

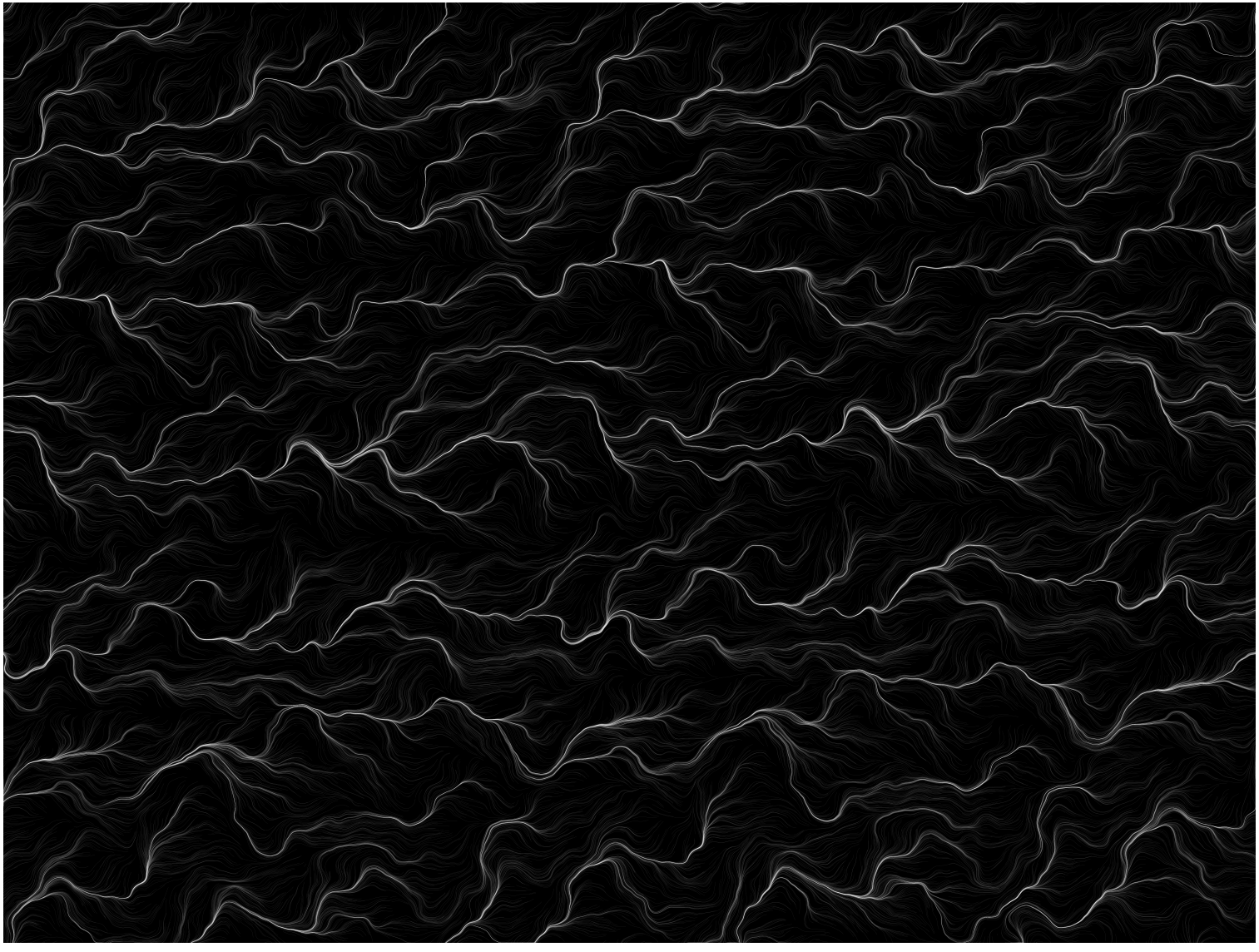
Technical Background: Physics, Electrical Engineering
Need image here!

About Me

Turn **science** into **art**.

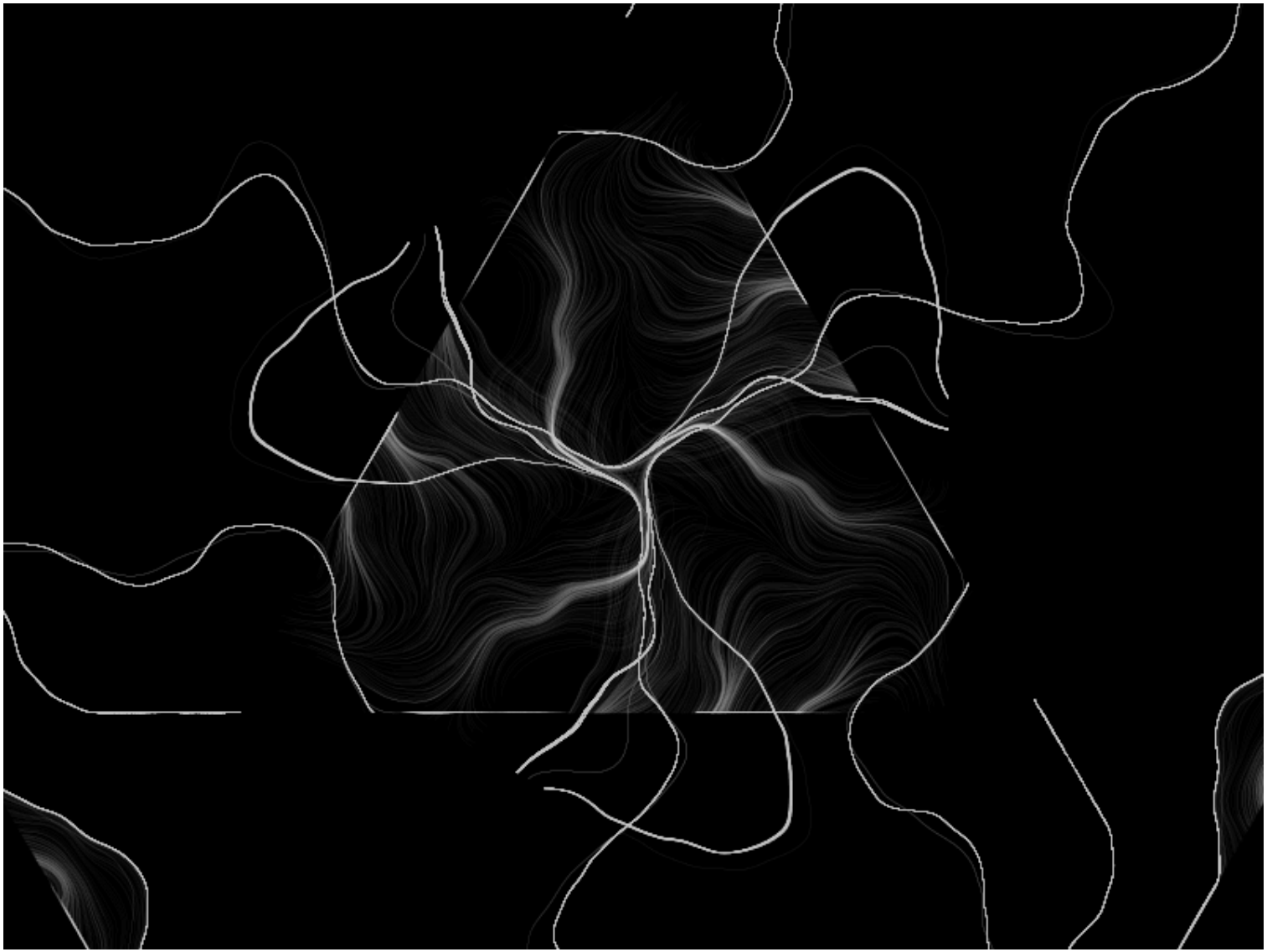
My Work

Randomness/Complexity



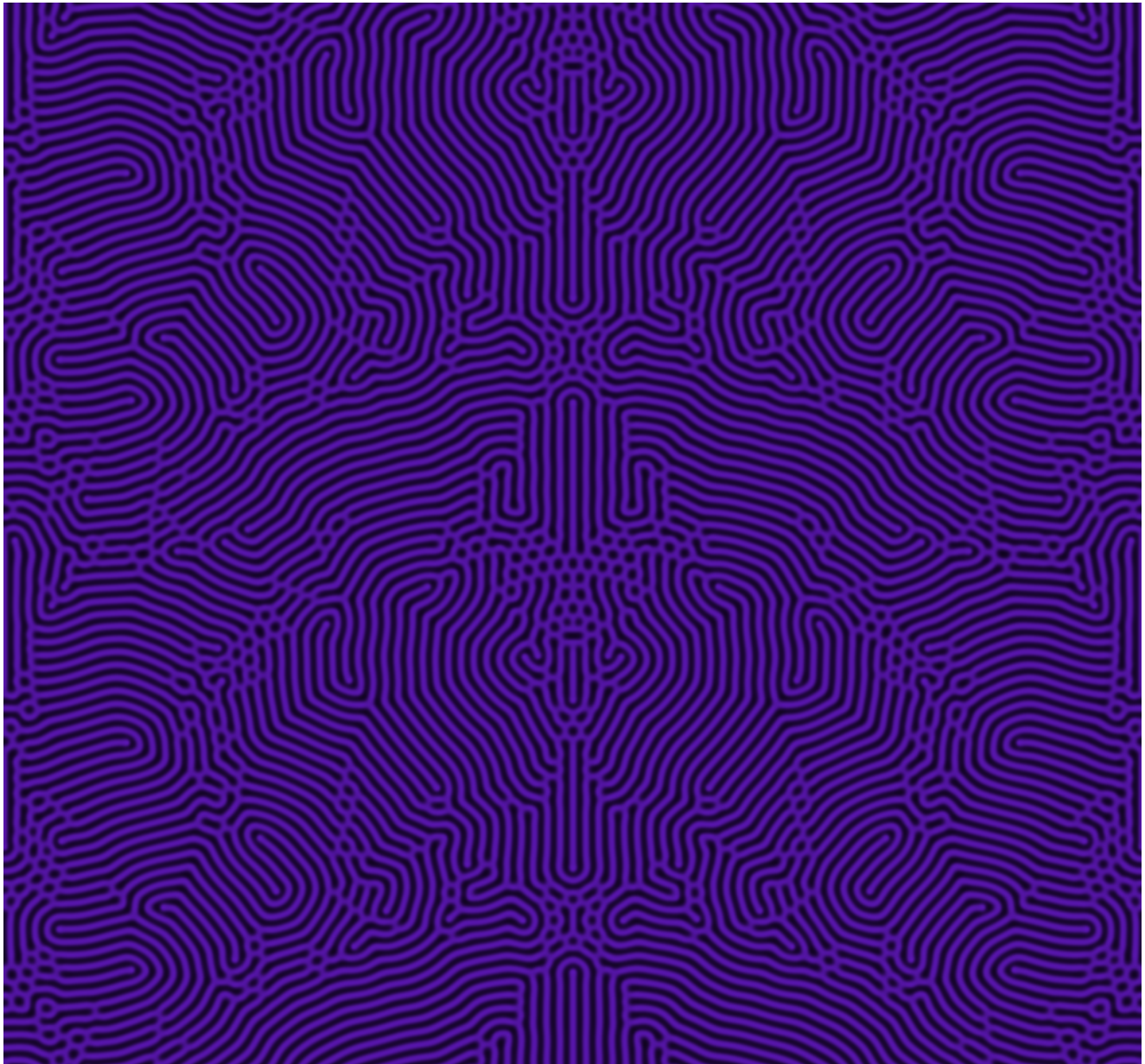
My Work

Geometry



My Work

Physical Phenomena



Introductions

- Name
- What do you want to get out of this class?
-

The Plan

Friday

- Install Processing
- Download Course Materials
- Brief introduction to programming concepts
- Practice!

The Plan

Friday

- 10:00 – Control/Logic
- 12:00 – Lunch Break
- 13:00 – Project 1
- 15:00 – Project 2

Install Processing

`http://www.processing.org/download`

Download Processing 1.5.1 (**NOT** 2.0b3!)

Download Course Materials

`fixme`

Structure

Human language is flexible in terms of diction and syntax.

Structure

Human language is flexible in terms of diction and syntax.
Computer language is not.

Structure

Expression $\approx$ Phrase

Structure

Expression $\approx$ Phrase

```
2 + 2
((3 * 4) + 2)
1< 2
```

Structure

Statement $\approx$ Sentence

Structure

Statement $\approx$ Sentence

```
size(500,250*2);
int x = 1;
```

Structure

Functions $\approx$ Paragraphs

Structure

Functions $\approx$ Paragraphs

```
background(0);
size(200,200);
```

Structure

Capitalization matters

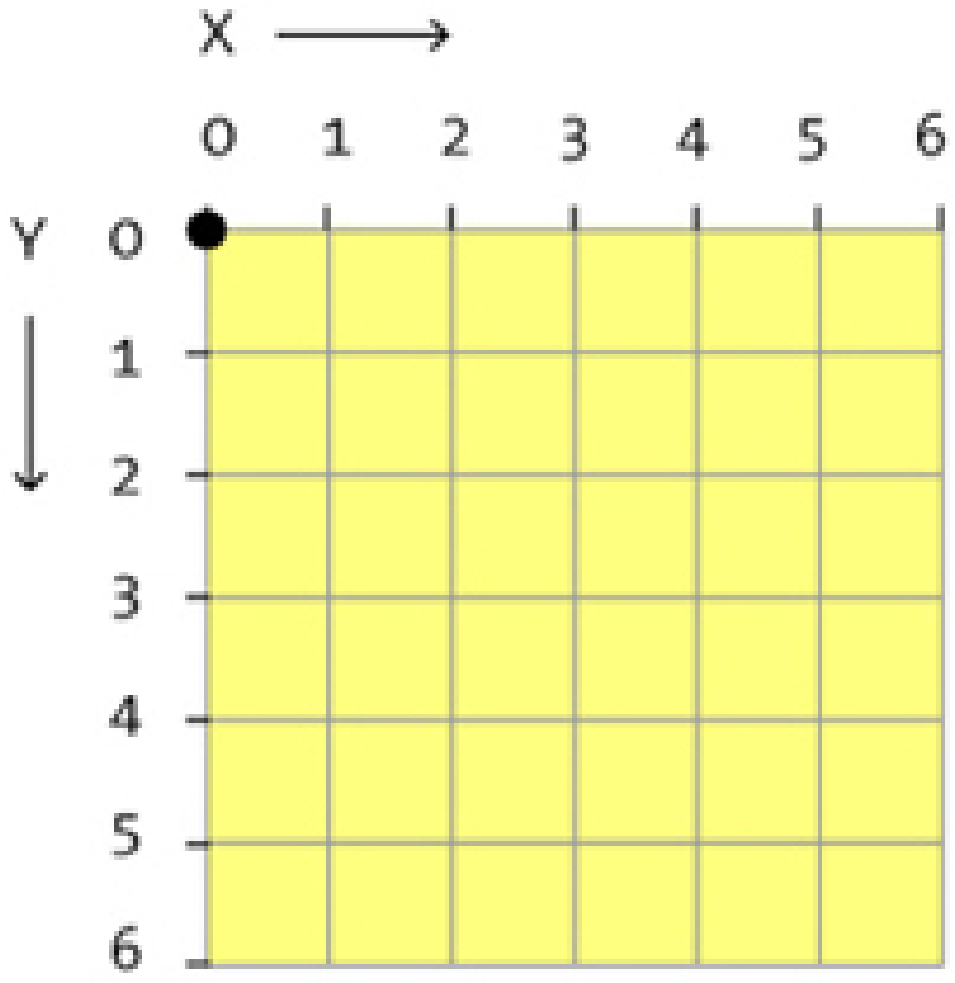
Structure

Capitalization matters
White Space doesn't

Structure

Capitalization matters
White Space doesn't
// creates a comment

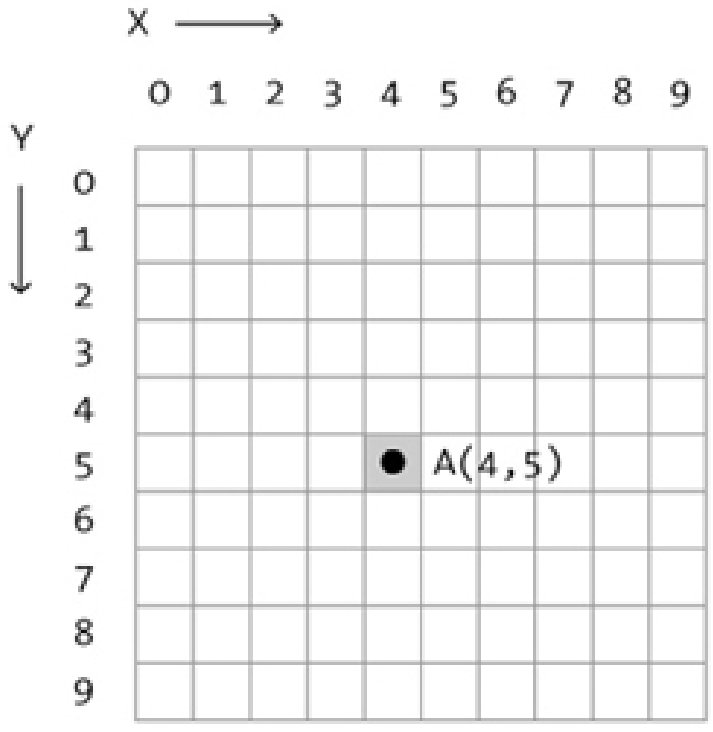
Primitives



Primitives

```
point(x position, y position)
```

Primitives




```
point(x,y);
```

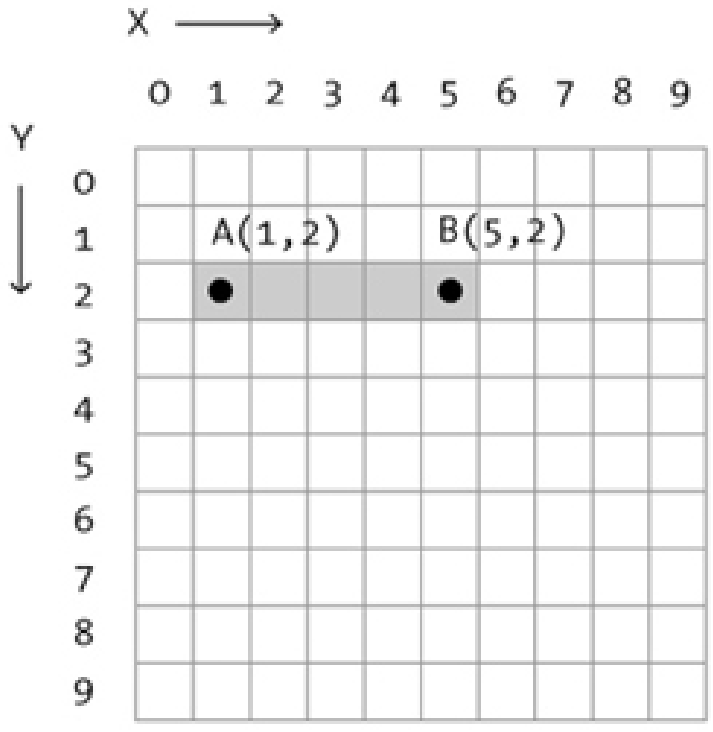
Example:

```
A(4,5);
```

Primitives

```
line(first x, first y, second x, second y)
```

Primitives




```
line(x1,y1,x2,y2);
```

Point A Point B

Example:

```
line(1,2,5,2);
```

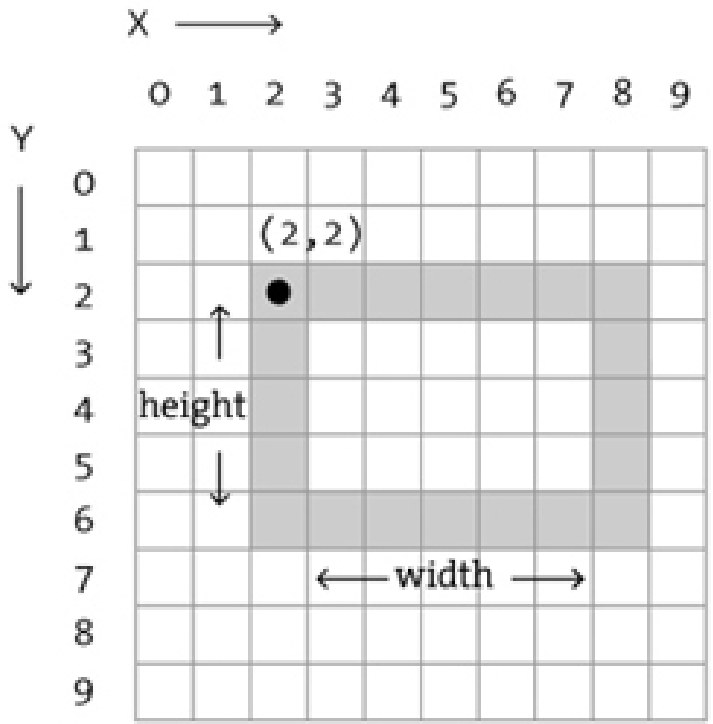
Primitives

Shapes

```
shape(center in x, center in y, size in x, size in
y)
```

Primitives

Rectangle



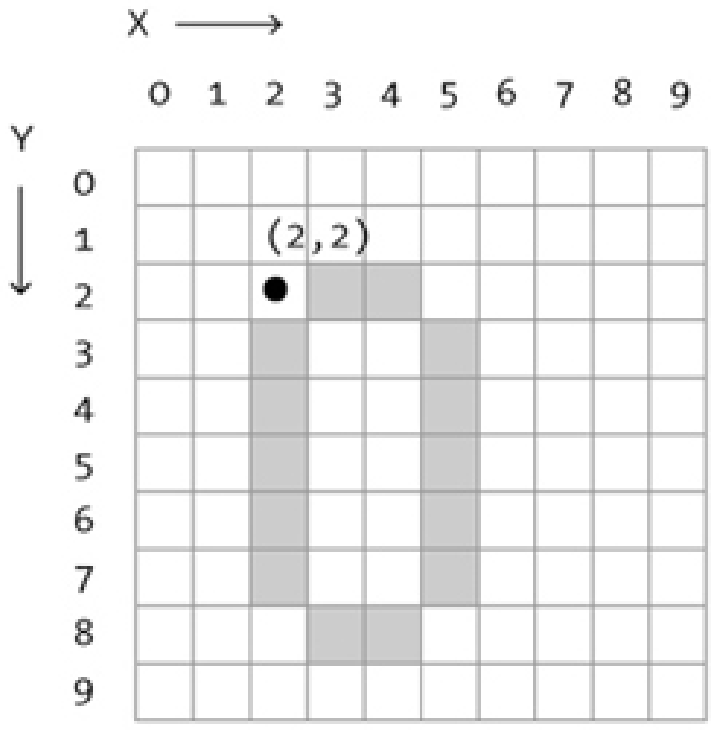

```
rect(x,y,width,height);
```

Example:

```
rect(2,2,7,5);
```

Primitives

Ellipse



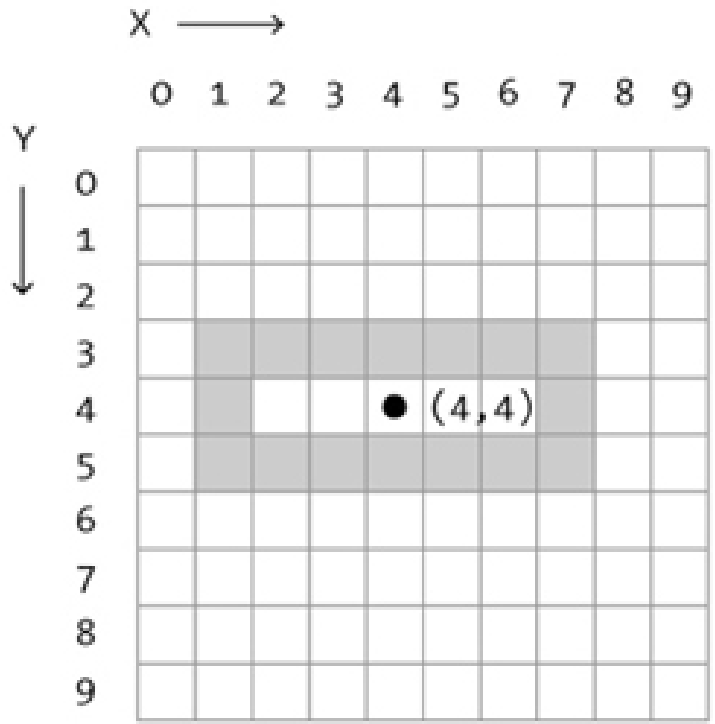

```
ellipseMode(CORNER);
ellipse(x,y,width,height);
```

Example:

```
ellipseMode(CORNER);
ellipse(2,2,4,7);
```

Primitives

Center Mode




```
rectMode(CENTER);
rect(x,y,width,height);
```

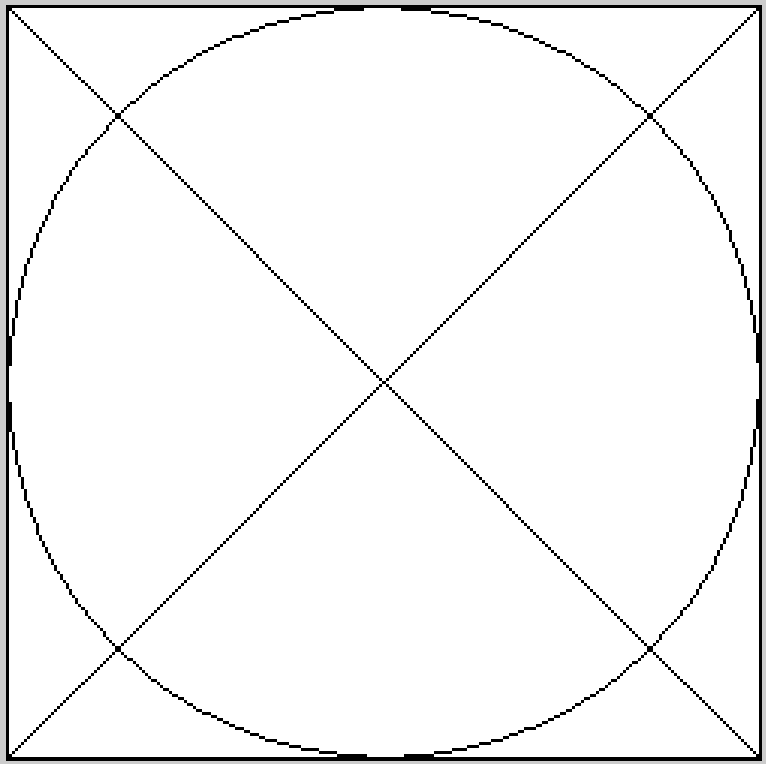
Example:

```
rectMode(CENTER);
rect(4,4,7,3);
```

Primitives

Try it!

Primitives



Colors

```
fill(greyscale)
```

Colors

```
fill(greyscale,alpha)
```

Colors

```
fill(red,blue,green)
```

Colors

```
fill(red,blue,green,alpha)
```

Colors

Related Functions

```
stroke()
noStroke()
noFill()
```

Colors

Try it!

Data Types

Try doubling the window size with `size()`.

Data Types

Try doubling the window size with `size()`.
The shapes aren't centered anymore!

Data Types

Variables are used to **store** data so that it can be **reused** easily.

Data Types

I

nitialize variable Assign value to variable

```
int x;
x = 5;
```

Data Types

- `int`
- `float`
- `boolean`

Data Types

Back to Processing!

Data Types

Draw an shape collage that scales with the window size.