Fundamentals of Physical Computing

Instructor: Rob Faludi

Plan for Today

- Lab Review
- Creative Analog Devices
- Math Review
- ~~Affordances~~
- Processing
- Readings & Assignments

Lab Review

Creative Analog Device Results

Math

Computer Math

- Binary, Decimal, Octal, Hexadecimal
    - Why?
        - Understanding computers
        - Programming helper, communications essential
    - What?
        - It’s all notation

Decimal

- Place system
- Powers
- Adding and carries
- Finger counting, but is that base 10?

Binary

- Place system
- Notation: %010 010b 0b10
- Powers
- Adding and carries
- Finger counting!

Octal

- Place system
- Notation **0**73
- Powers
- Adding and carries
- Finger counting, not really

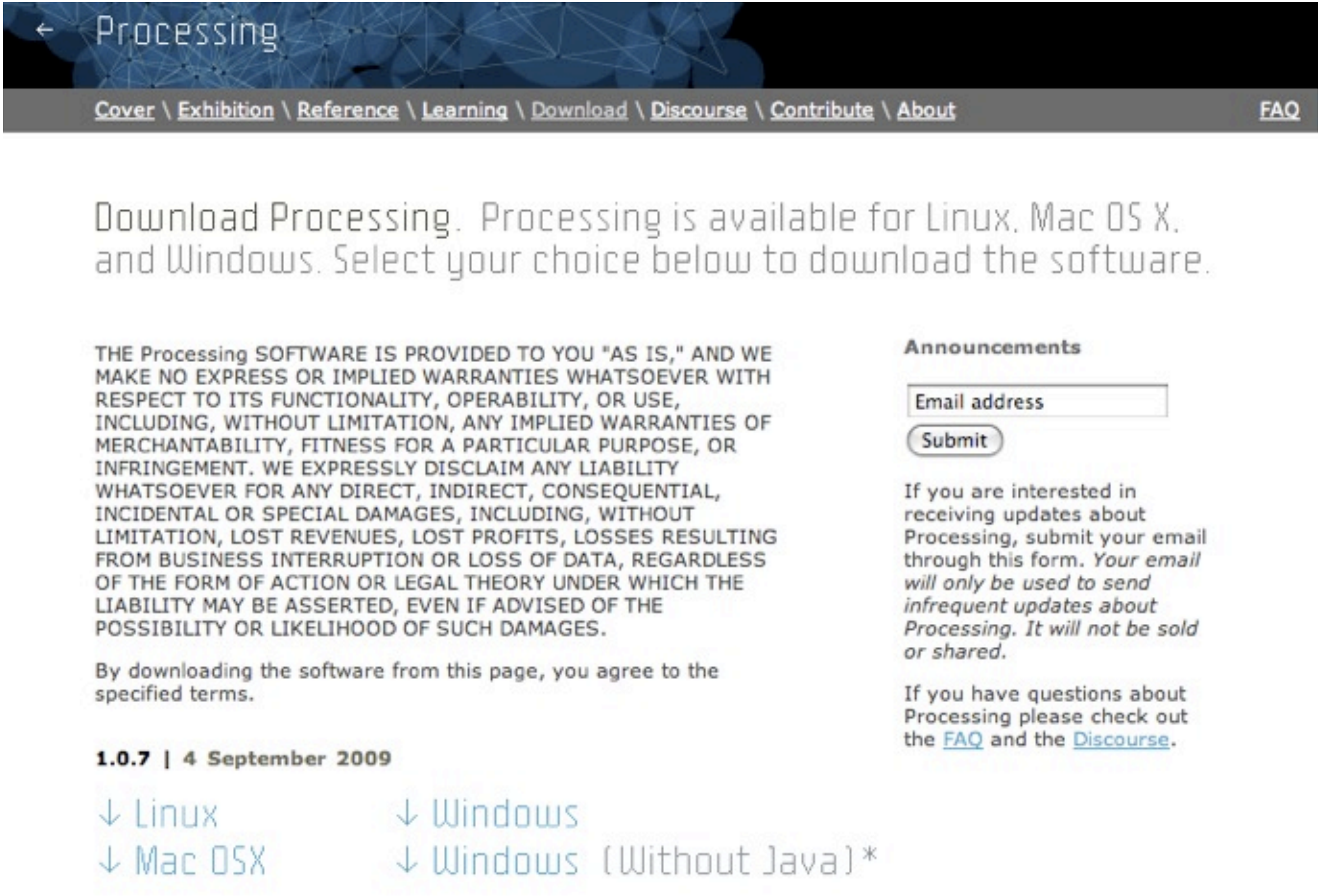
Hexadecimal

- Place system
- Notation: extra digits, 0x10, #FFFFFF
- Powers
- Adding and carries
- Finger counting?
- Switches yes: 0xFF = 1111 1111 and 0x3C = 0011 1100

Programming

Processing

Downloading Processing


← Processing
Cover \ Exhibition \ Reference \ Learning \ Download \ Discourse \ Contribute \ About
FAQ

Download Processing. Processing is available for Linux, Mac OS X, and Windows. Select your choice below to download the software.

THE Processing SOFTWARE IS PROVIDED TO YOU "AS IS," AND WE MAKE NO EXPRESS OR IMPLIED WARRANTIES WHATSOEVER WITH RESPECT TO ITS FUNCTIONALITY, OPERABILITY, OR USE, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR INFRINGEMENT. WE EXPRESSLY DISCLAIM ANY LIABILITY WHATSOEVER FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, INCIDENTAL OR SPECIAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST REVENUES, LOST PROFITS, LOSSES RESULTING FROM BUSINESS INTERRUPTION OR LOSS OF DATA, REGARDLESS OF THE FORM OF ACTION OR LEGAL THEORY UNDER WHICH THE LIABILITY MAY BE ASSERTED, EVEN IF ADVISED OF THE POSSIBILITY OR LIKELIHOOD OF SUCH DAMAGES.

By downloading the software from this page, you agree to the specified terms.

1.0.7 | 4 September 2009

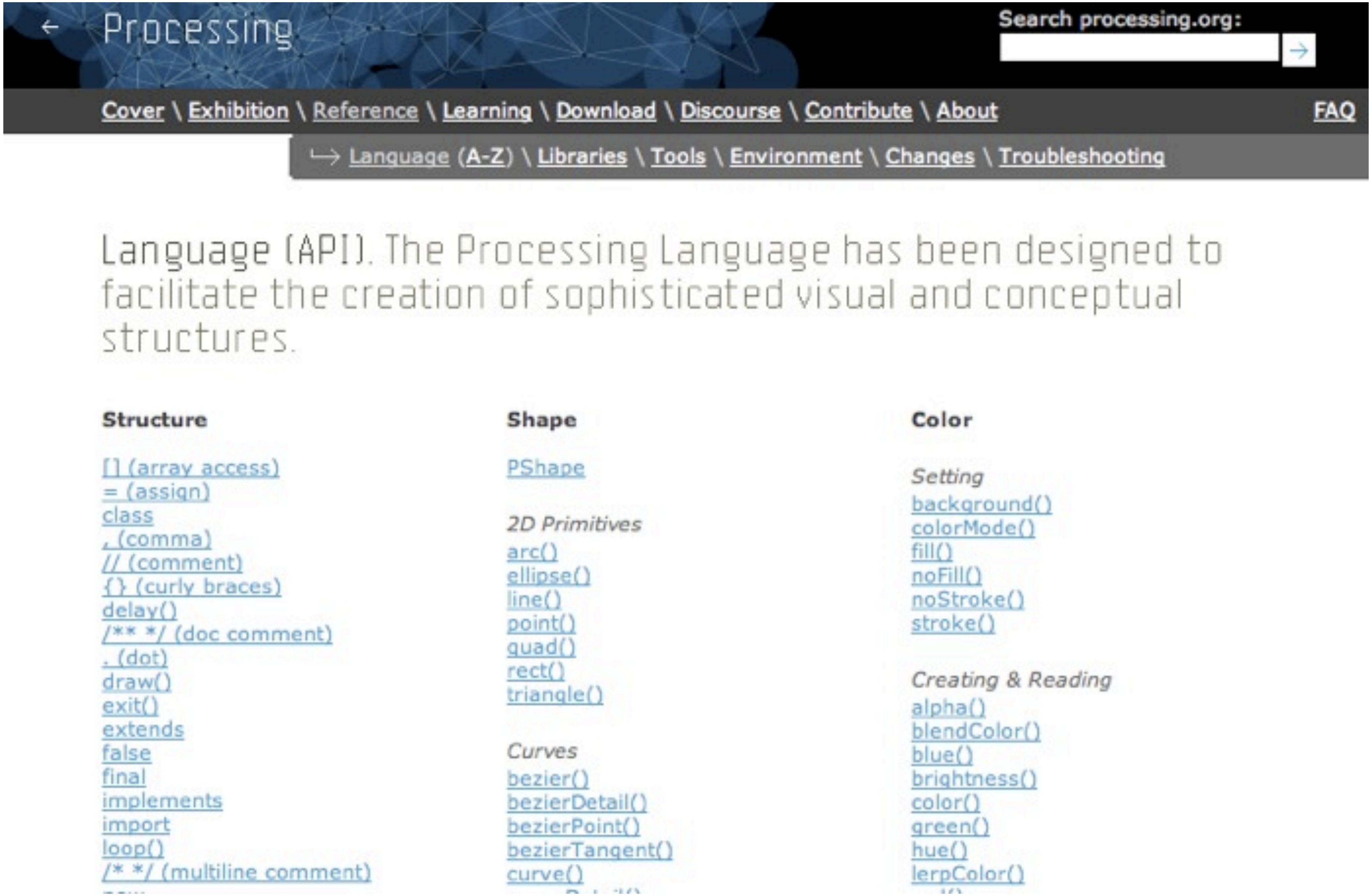
↓ Linux
↓ Mac OSX
↓ Windows
↓ Windows (Without Java)*

Announcements

If you are interested in receiving updates about Processing, submit your email through this form. *Your email will only be used to send infrequent updates about Processing. It will not be sold or shared.*

If you have questions about Processing please check out the FAQ and the Discourse.

Processing Web Site



← Processing

Search processing.org: →

Cover \ Exhibition \ Reference \ Learning \ Download \ Discourse \ Contribute \ About

FAQ

↳ Language (A-Z) \ Libraries \ Tools \ Environment \ Changes \ Troubleshooting

Language (API). The Processing Language has been designed to facilitate the creation of sophisticated visual and conceptual structures.

Structure

[] (array access)
= (assign)
class
, (comma)
// (comment)
{} (curly braces)
delay()
/** */ (doc comment)
. (dot)
draw()
exit()
extends
false
final
implements
import
loop()
/* */ (multiline comment)

Shape

PShape

2D Primitives

arc()
ellipse()
line()
point()
quad()
rect()
triangle()

Curves

bezier()
bezierDetail()
bezierPoint()
bezierTangent()
curve()

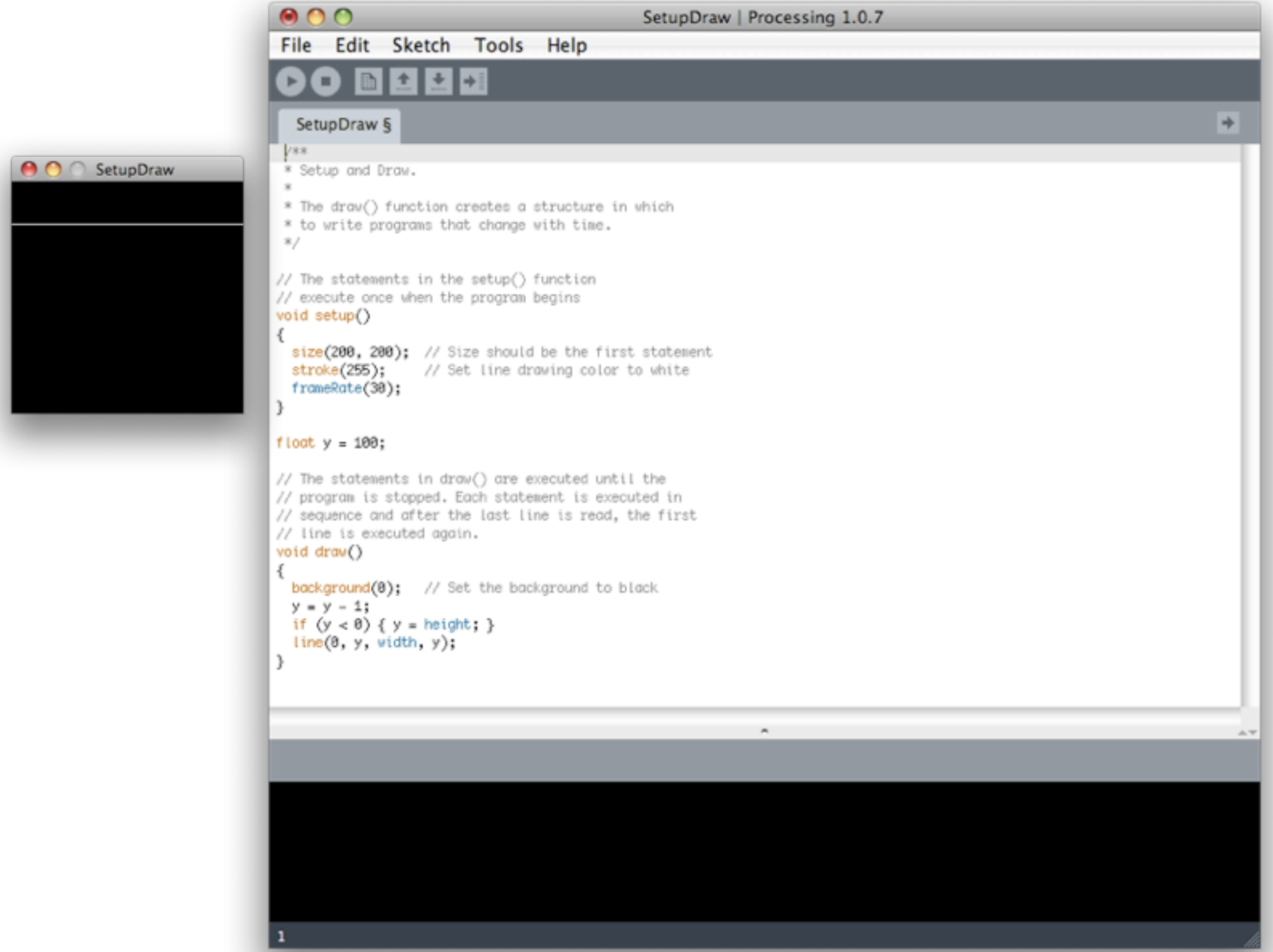
Color

Setting

background()
colorMode()
fill()
noFill()
noStroke()
stroke()

Creating & Reading

alpha()
blendColor()
blue()
brightness()
color()
green()
hue()
lerpColor()



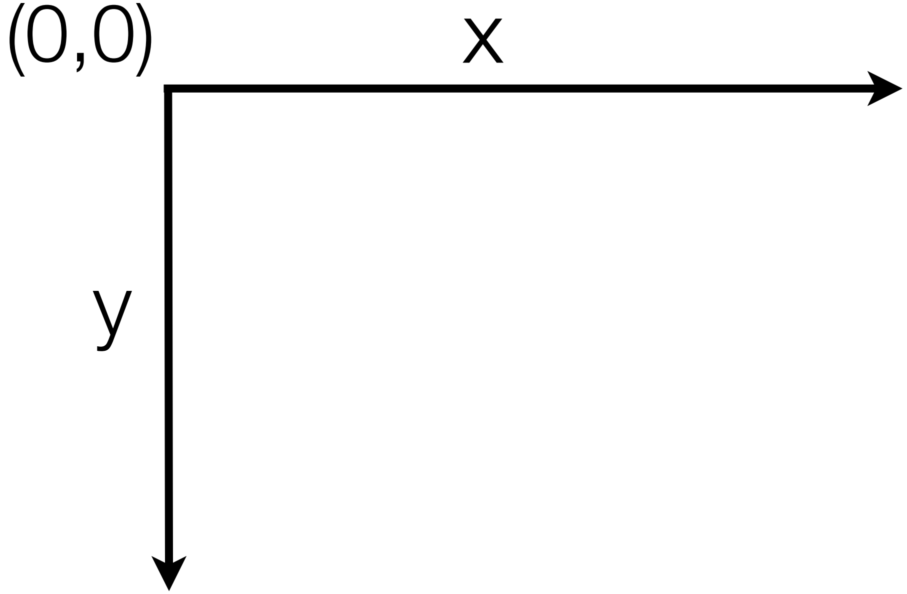
```
/**
 * Setup and Draw.
 *
 * The draw() function creates a structure in which
 * to write programs that change with time.
 */

// The statements in the setup() function
// execute once when the program begins
void setup()
{
  size(200, 200);  // Size should be the first statement
  stroke(255);     // Set line drawing color to white
  frameRate(30);
}

float y = 100;

// The statements in draw() are executed until the
// program is stopped. Each statement is executed in
// sequence and after the last line is read, the first
// line is executed again.
void draw()
{
  background(0);   // Set the background to black
  y = y - 1;
  if (y < 0) { y = height; }
  line(0, y, width, y);
}
```

Coordinate System

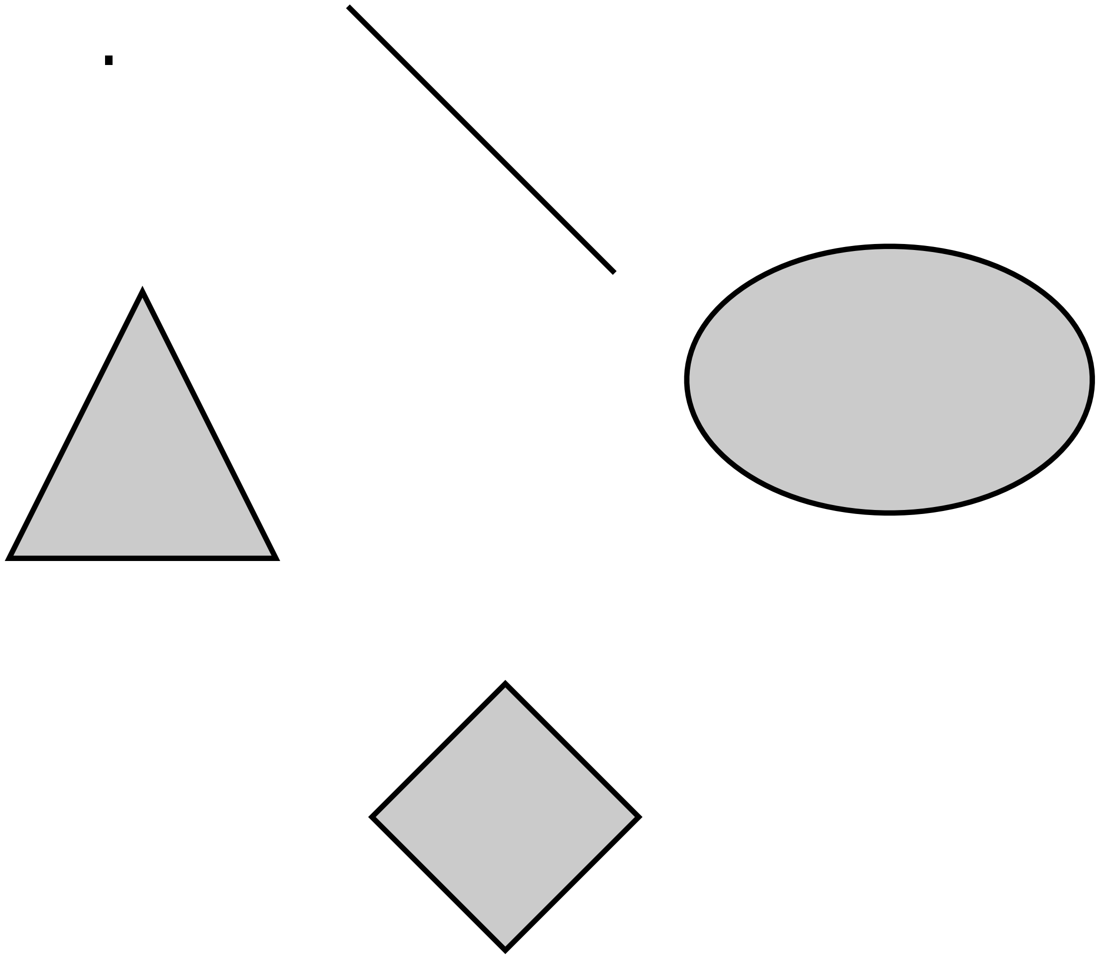


(5, 100) ?

(75,10) ?

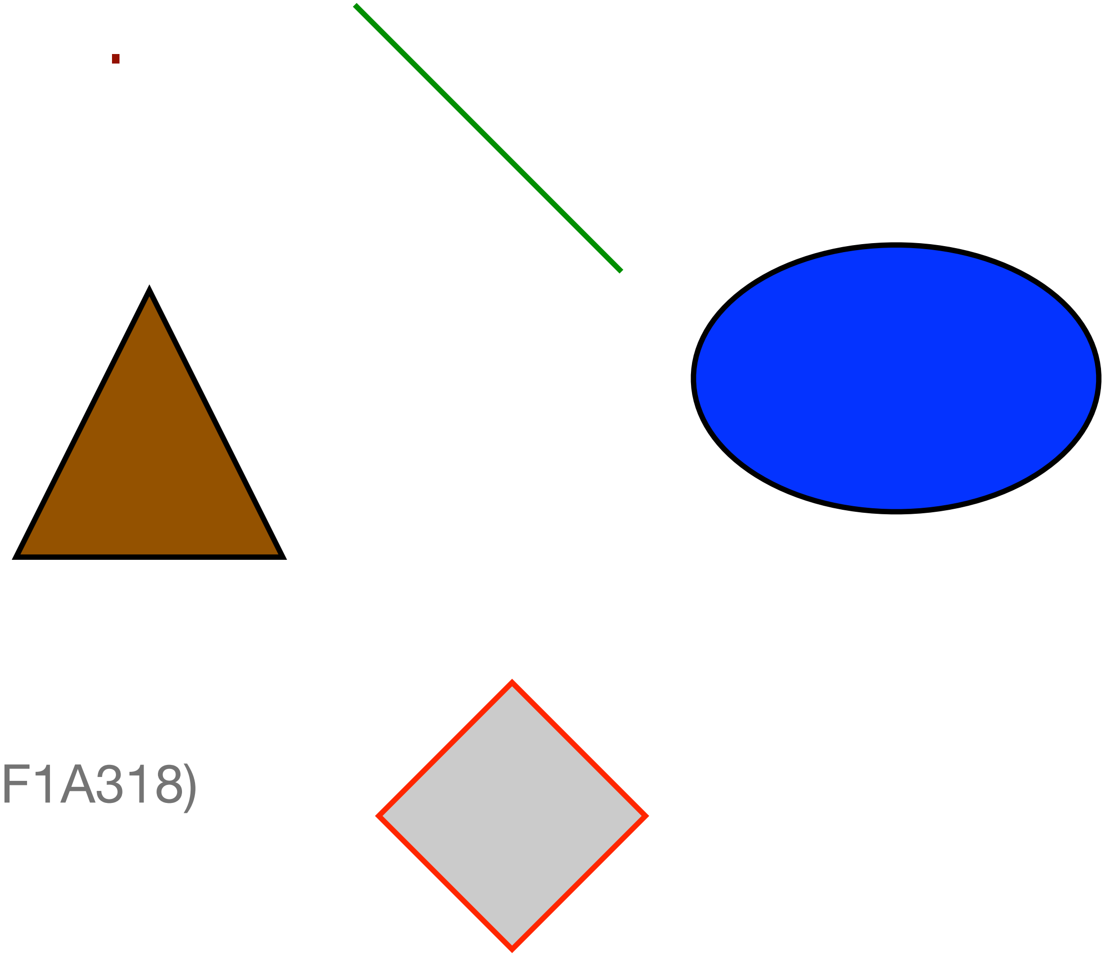
Drawing Shapes

- Primitives:
    - point
    - line
    - triangle
    - quad
    - rect
    - ellipse



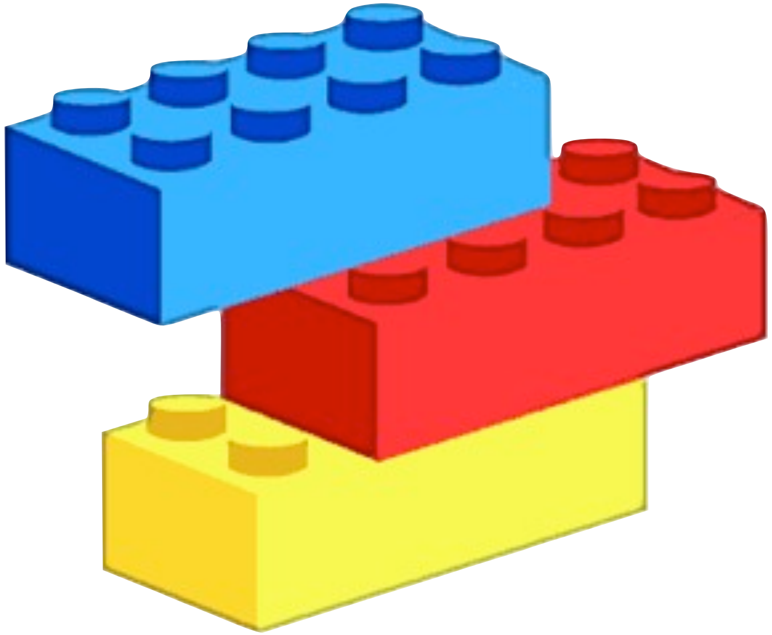
Colors

- fill( )
- stroke( )
- background( )
- (32)   or (213, 5,118) or (#F1A318)

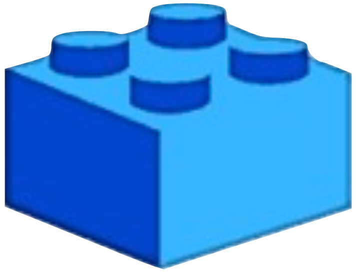


Structure

- setup( )
- draw( )



- functions and loops use curly brackets { }
- lines end with semicolons ;



Variables

- int
- float
- color
- boolean
- mouseX
- mouseY
- mousePressed

Conditionals

- if (mouseX > 50)

- if (mousePressed)

- if (mouseX > 50 && mouseY > 4)

Comments and Console

- // comment
- /* comment */
- print & println

Readings and Assignments

- Readings
    - Physical Computing, chapter 5
- Assignment
    - Programming Lab