

# creativity & computation lab

PGTE 5250

TUESDAY 3:50-6:30 PM

RM 1202, 6 EAST 16TH ST

MFA DESIGN+TECHNOLOGY, PARSONS THE NEW SCHOOL FOR DESIGN

Liza Stark

[stare220@newschool.edu](mailto:stare220@newschool.edu)

[lizastark.com/cclabfall2012](http://lizastark.com/cclabfall2012)

# what is CC lab?

GIVE US TOOLS!

- Develop a **foundation** in the basics of computation
- Introduce you to new **tools** that you can use to realize/build your projects:
  - Survey three different programming languages
  - What is the best tool for the job?
- Practical, hands-on experience

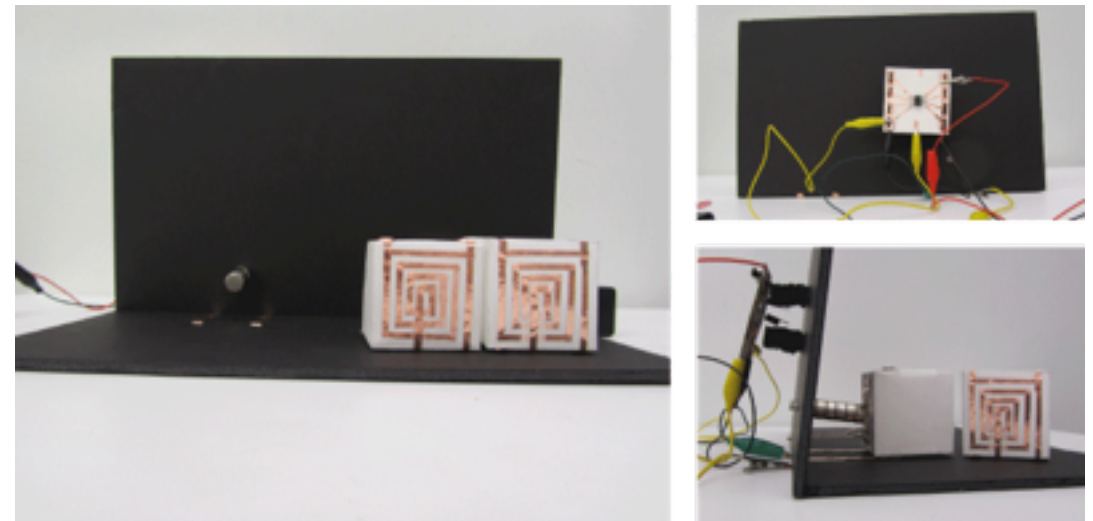
# what is CC lab NOT?

GIVE US TOOLS!

- Super in depth study of one programming language
- Becoming a hardcore coder
- Learning how to hack code without understanding it

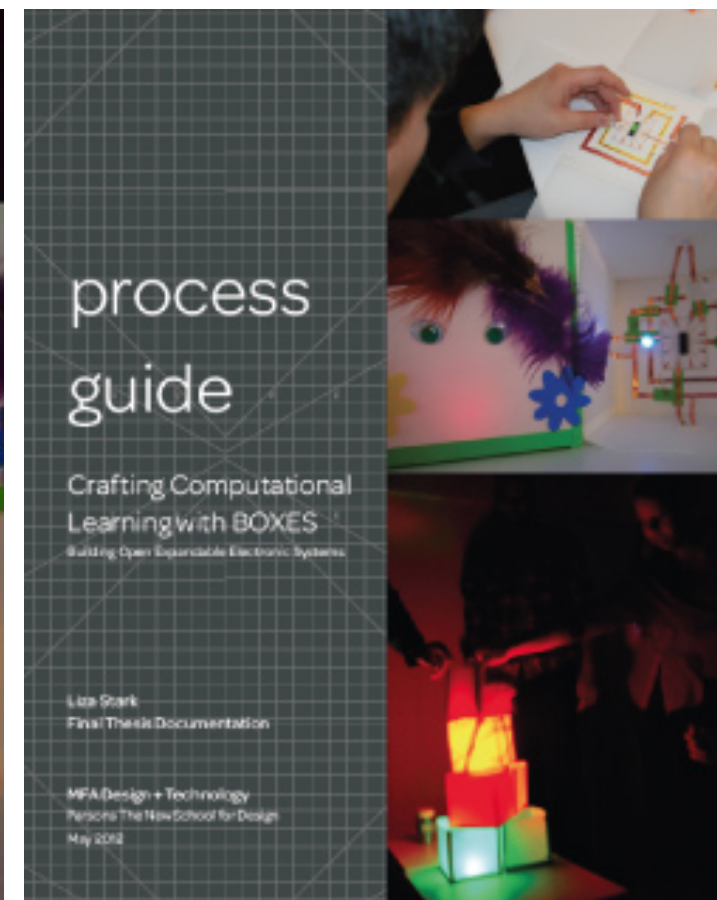
# introductions

WHO AM I?



# introductions

WHO AM I?



4 make the connection

Circuits are all about connection. Keeping the wrong things separate is just as important as connecting the right things.



GROUND RULE #1: Circuits are all about connection. Keeping the wrong things separate is just as important as connecting the right things.

explain it.

Like all systems, circuits are about flow. Electricity can't flow if it's not connected properly, and it won't be able to do its job if it's not connected to the right things.

The best way to make sure you have a good circuit is to make sure you have a good connection.

30 BOXES

6 let there be light

You can add a resistor to make sure the circuit recognizes an open or closed switch.



GROUND RULE #2: Circuits are all about connection. Keeping the wrong things separate is just as important as connecting the right things.

explain it.

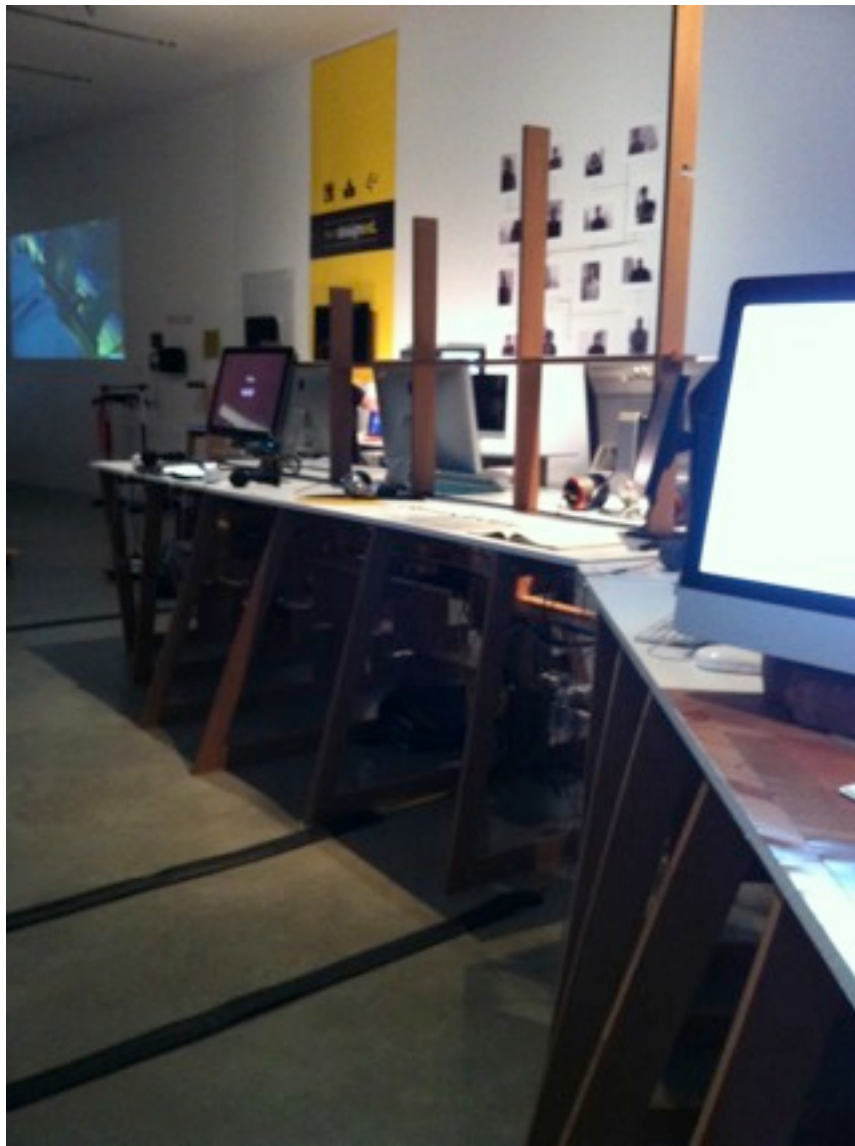
LED stands for Light Emitting Diode. It's a semiconductor that can only move in one way, so it's perfect for making sure you have a good connection.

40 BOXES



# introductions

WHO AM I?



# introductions

WHO AM I?

## My role as your professor

Guidance

Teach you how to teach yourself

Help you design and execute awesome projects

# introductions

WHO ARE YOU?

## Your role as a CC Lab student

Complete your assignments

Participate in class

\*\*\*Help your peers



# introductions

WHO ARE YOU?

Presentation time!

# bootcamp

LET'S AVOID BAD FLASHBACKS

What were your....

Triumphs?

Challenges?

Problem areas?

# syllabus

## OVERVIEW

What exactly are we going to be  
doing...

# let's review

BACK TO BASICS

## Some **big** questions

What is a programming language?

What tools are we learning?

//Which one is best for the project?

How do I study?

# what is a programming language?

BACK TO BASICS

A language used **to communicate instructions** to a computer.

Allows humans to communicate with machines.

Composed of **syntax** (form) and **semantics** (meaning).

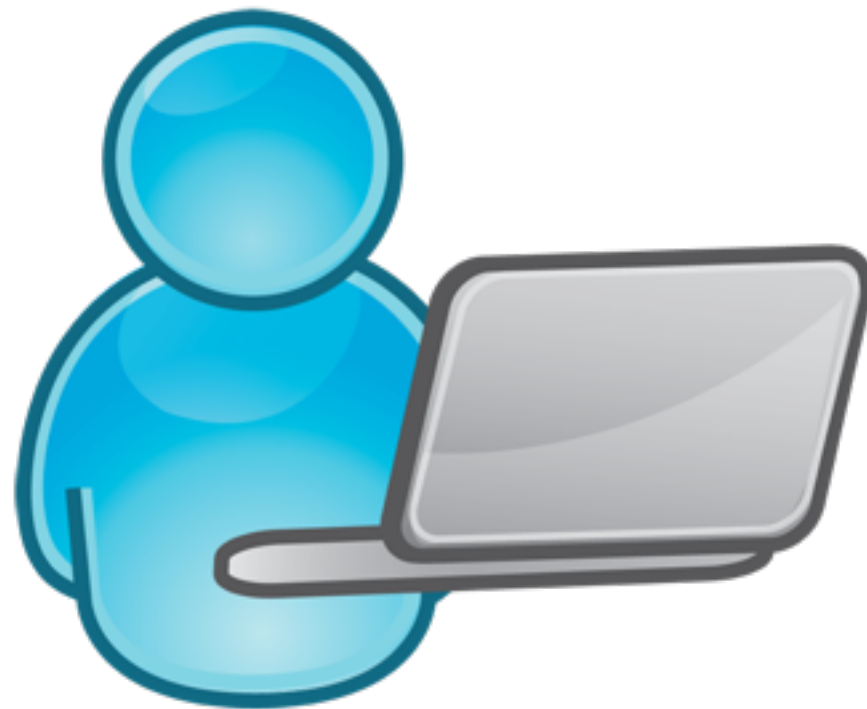
Performs computations or algorithms  
Controls external devices.

High/Low level languages.

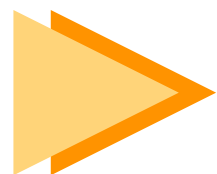


# what is a programming language?

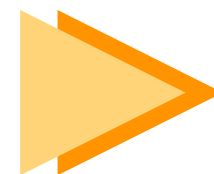
BACK TO BASICS



Hey computer!  
Draw me a  
rectangle at 56, 46  
with a width of 55  
and a height of 20!



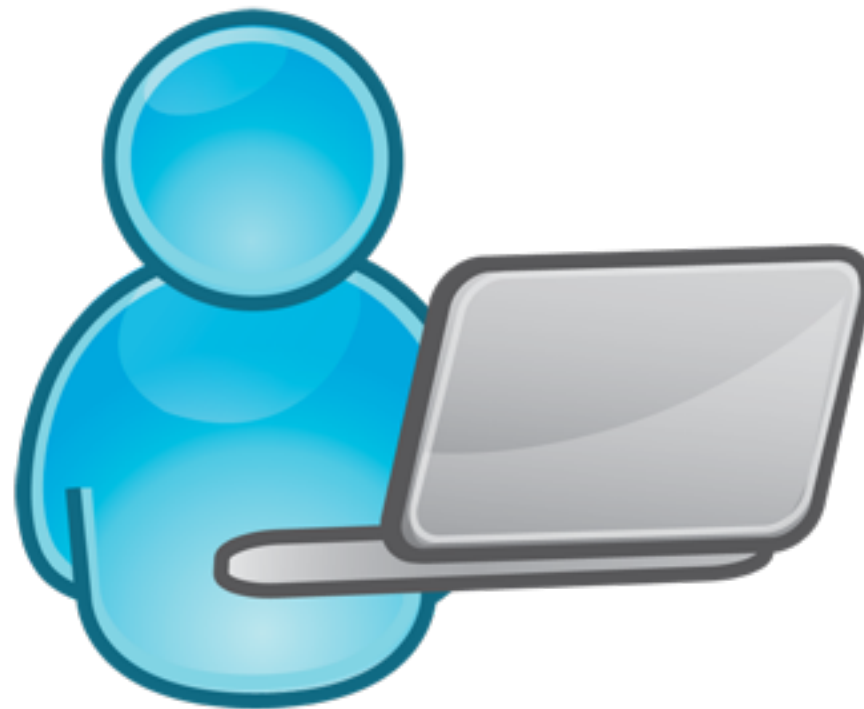
```
rect(56, 46, 55, 20);
```



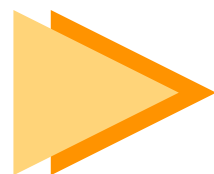
```
1001001 010001  
10000 10100...
```

# what is a programming language?

BACK TO BASICS



Hey computer!  
Draw me a  
rectangle at 56, 46  
with a width of 55  
and a height of 20!



```
rect(56, 46, 55, 20);
```



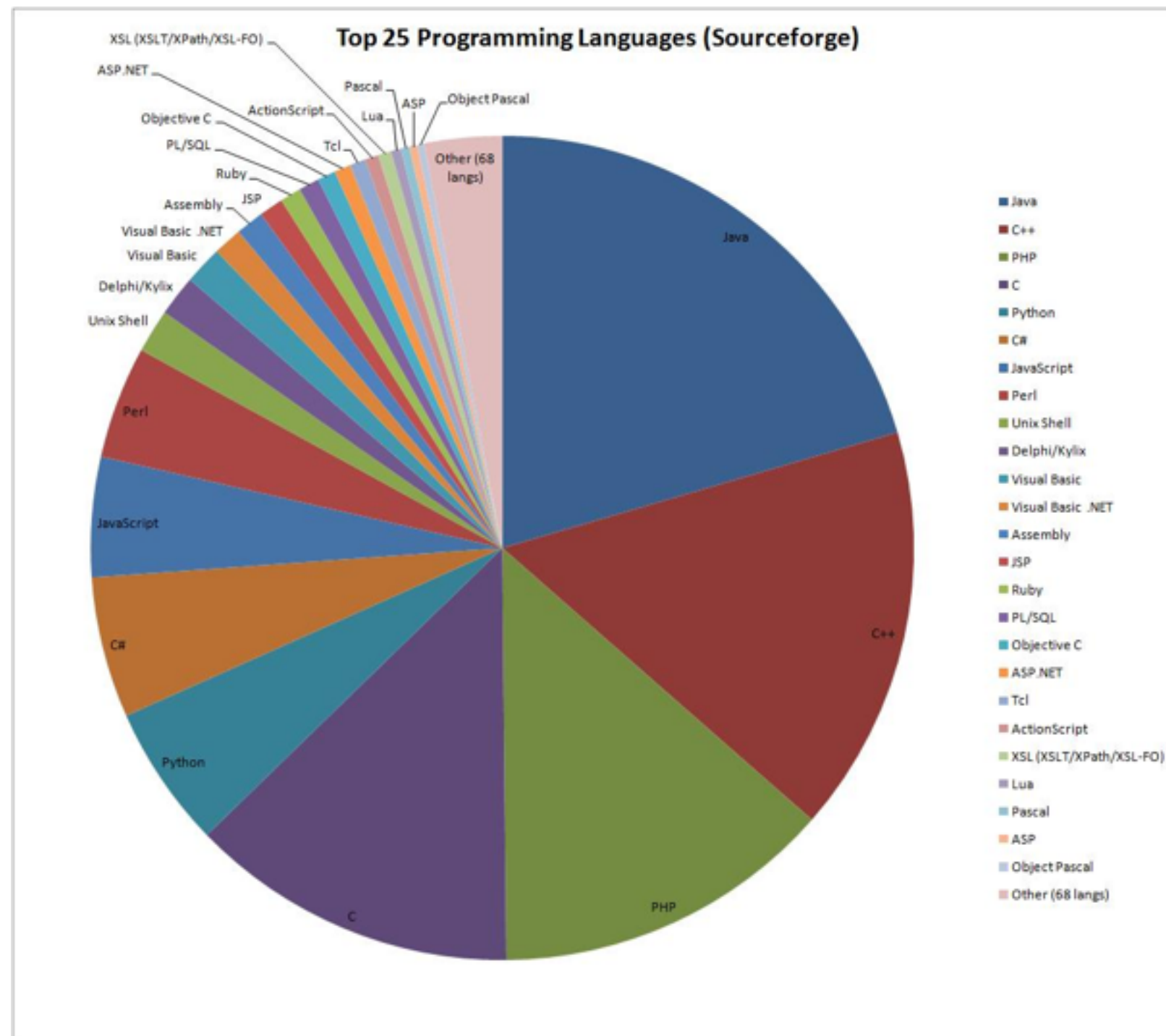
```
1001001 010001  
10000 10100...
```



Compiler (translator)

# what is a programming language?

THERE ARE SO FREAKING MANY!



# what tools are we learning?

WHICH ONE IS BEST FOR THE PROJECT?

## Processing



Built on top of Java programming language

//Processing is actually a programming environment

Great tool to begin learning the basics of programming.

Widely used for education - LOTS of resources.

Good for quick prototyping.

Easy integration with Arduino.

# what tools are we learning?

WHICH ONE IS BEST FOR THE PROJECT?

## Arduino

//This is my jam.



Modeled on the Processing environment and uses simplified C++ commands and functions

Open source hardware/software.

Great tool to for rapid prototyping physical interfaces.

You can basically make anything with it.

Easy integration with Processing and oF.



# what tools are we learning?

WHICH ONE IS BEST FOR THE PROJECT?

## openFrameworks



Framework and set of libraries written in C++.

//You are actually writing C++

For more than the beginner, but more control over everything.

Can do things that Processing cannot.

Better for large scale projects.

Integrates with Arduino.



Display on interwebs

Runs easily on lots of computers

Quick prototypes

Interface with  
Arduino

Use openCV

Work in video or 3D graphics

iOS or Android

Anything with  
physical interaction  
or interfaces

Gathering real  
world/time  
data with sensors



a comparison

a.k.a. What is the **best** tool  
for the project?

# how do i study?

JUST LIKE LEARNING ANOTHER LANGUAGE

## Programming language.

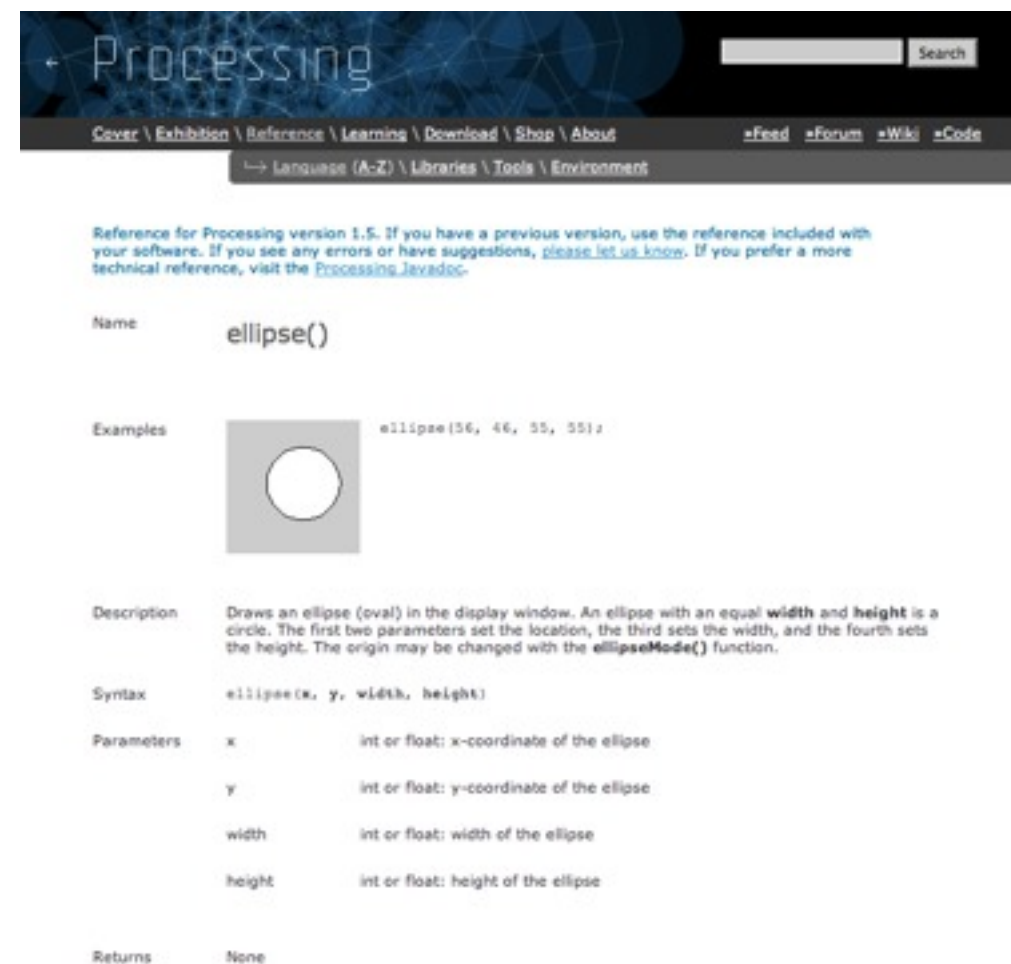
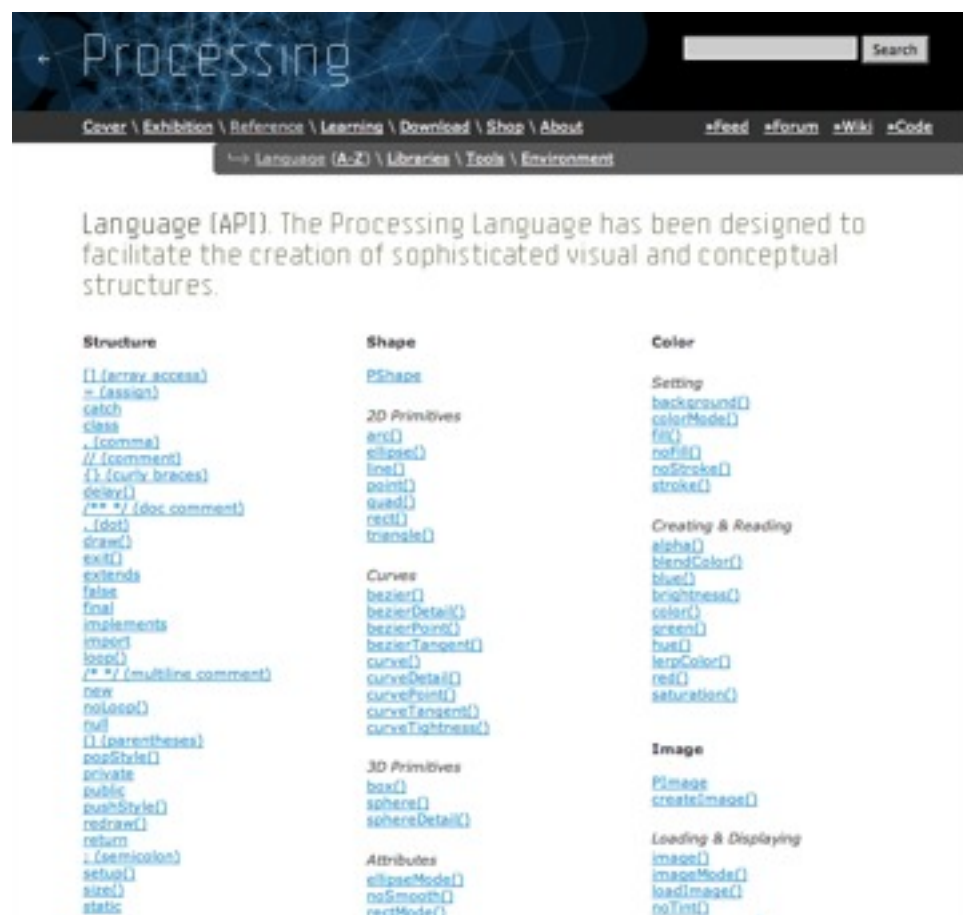
You need to use it frequently.

This is **not** about ~~memorizing~~.

# how do i study?

JUST LIKE LEARNING ANOTHER LANGUAGE

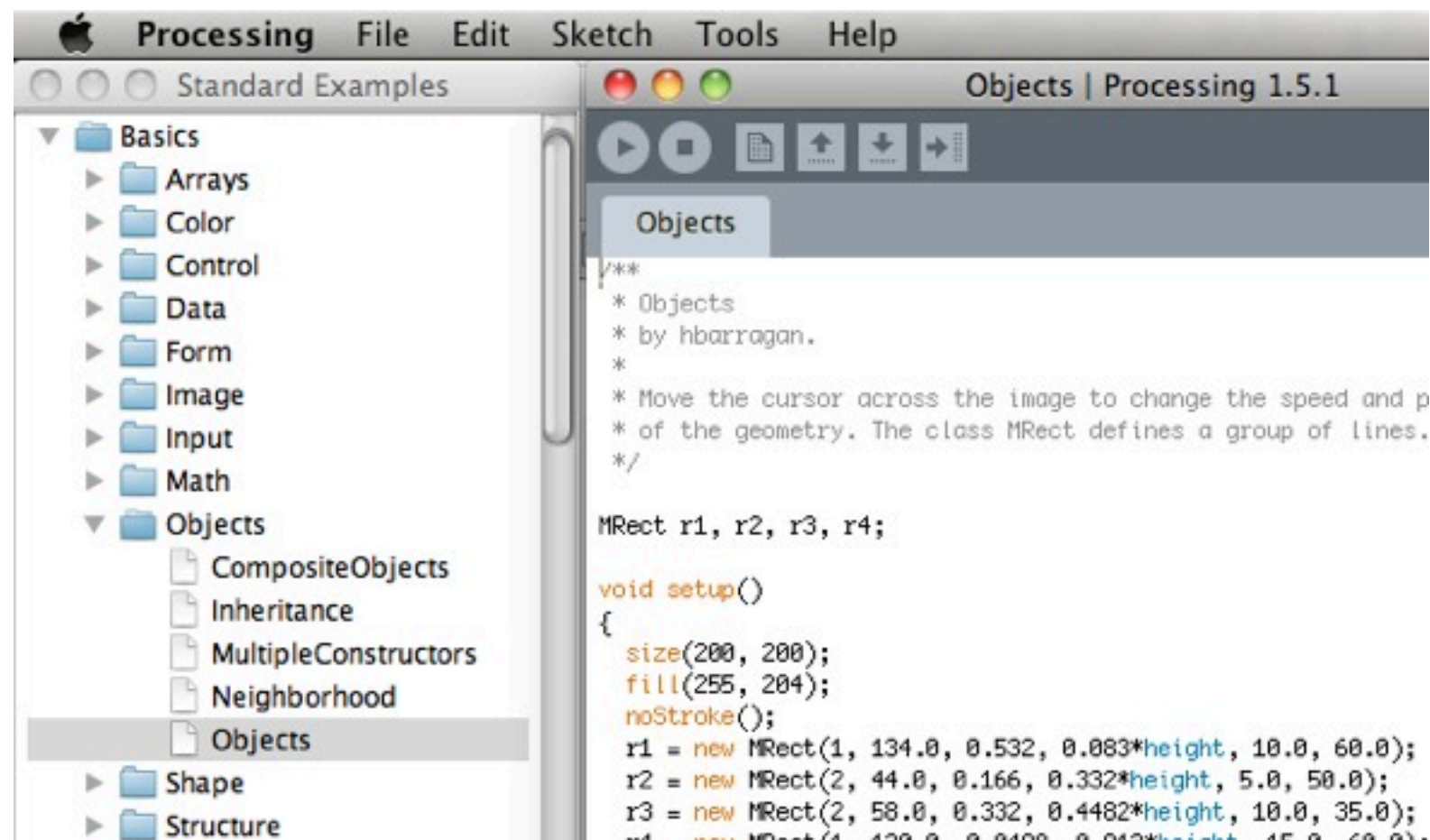
1. **Reference** – every programming language has a documented reference. This is just like a dictionary.



# how do i study?

JUST LIKE LEARNING ANOTHER LANGUAGE

2. Tinker with **examples** and **sample code** - these are a great starting point! You can run examples, examine the code, and edit different parts to see what happens.

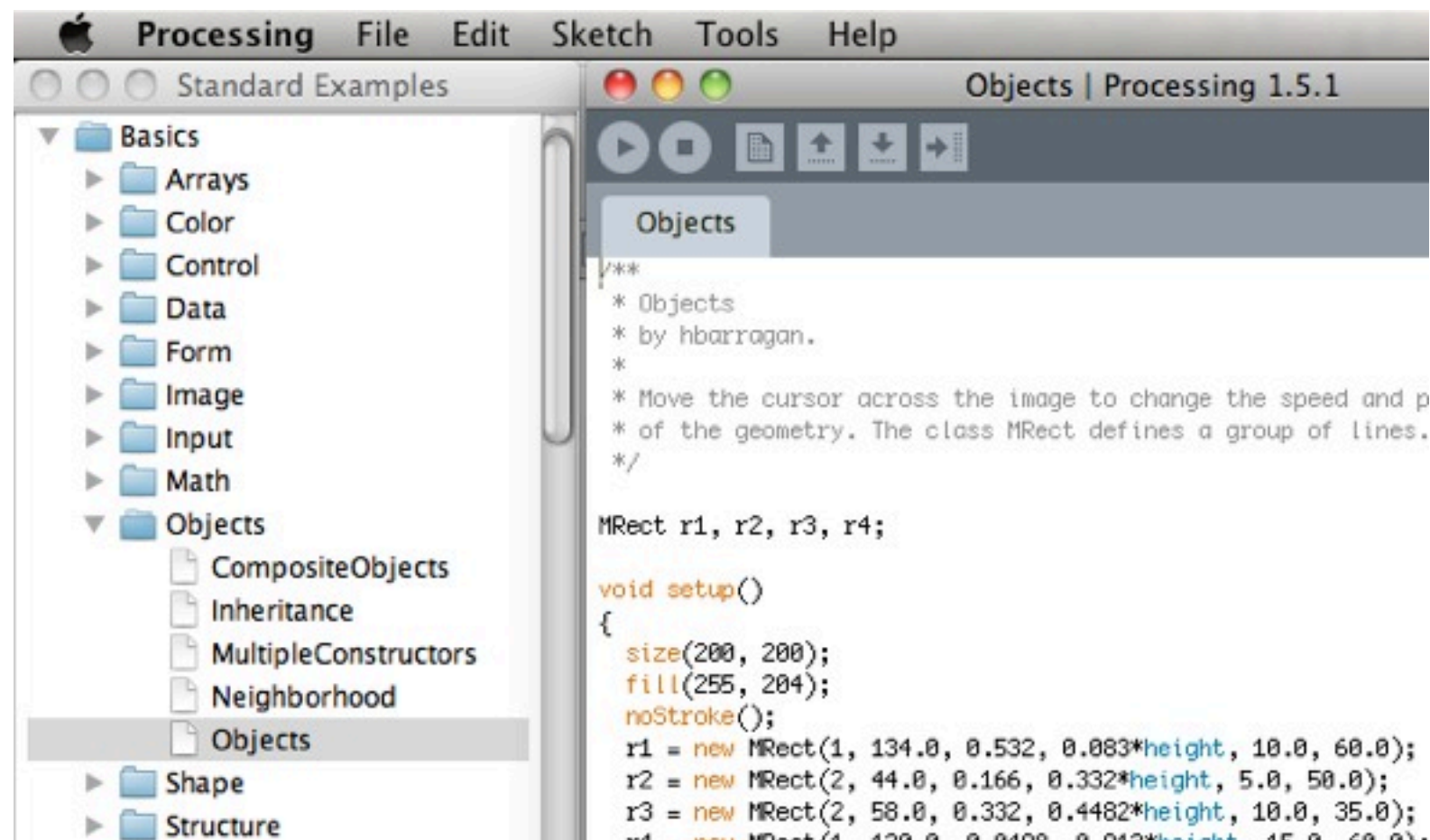




# how do i study?

JUST LIKE LEARNING ANOTHER LANGUAGE

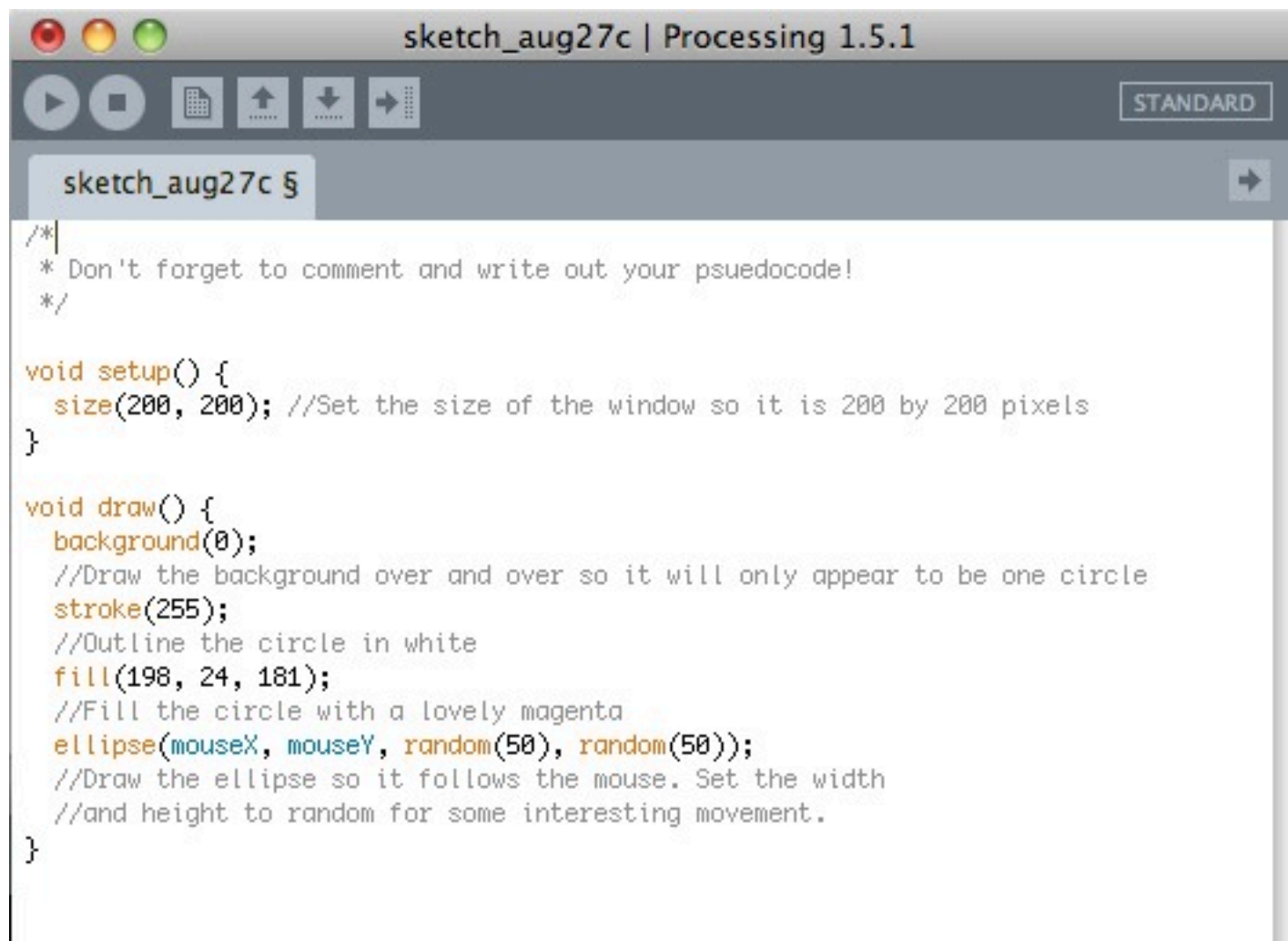
Don't forget the examples in books - you may have to actually copy them, but this is good practice!



# how do i study?

JUST LIKE LEARNING ANOTHER LANGUAGE

3. **Commenting** and **pseudocode** - not only are these requirements for the class, they are super helpful for structuring your code or when you get stuck in a problem.



```
sketch_aug27c | Processing 1.5.1
STANDARD
sketch_aug27c §
/*
 * Don't forget to comment and write out your psuedocode!
 */

void setup() {
  size(200, 200); //Set the size of the window so it is 200 by 200 pixels
}

void draw() {
  background(0);
  //Draw the background over and over so it will only appear to be one circle
  stroke(255);
  //Outline the circle in white
  fill(198, 24, 181);
  //Fill the circle with a lovely magenta
  ellipse(mouseX, mouseY, random(50), random(50));
  //Draw the ellipse so it follows the mouse. Set the width
  //and height to random for some interesting movement.
}
```

# how do i study?

JUST LIKE LEARNING ANOTHER LANGUAGE

4. **Get inspired!** Projects that inspire you will motivate you.



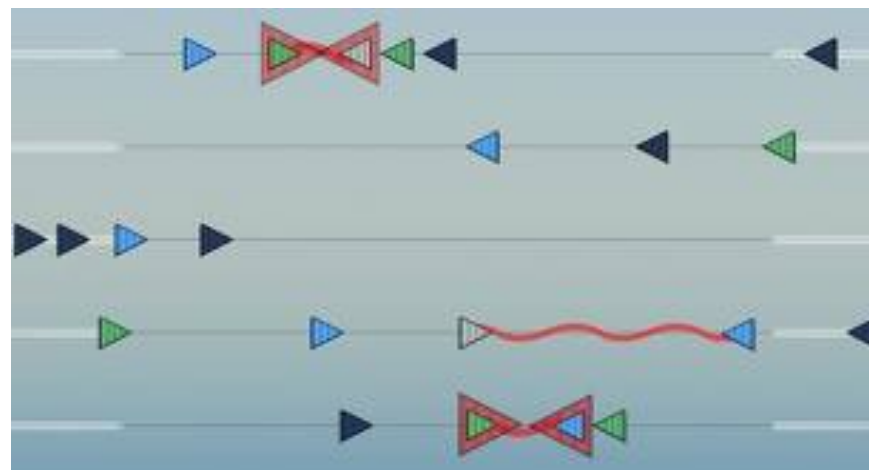
[openprocessing.org](http://openprocessing.org)



# how do i study?

JUST LIKE LEARNING ANOTHER LANGUAGE

## 4. Every complex project started as something small...



Everything complex project started as something small!

# recalibrate

IT'S NOT A TEST I SWEAR

## Survey time!

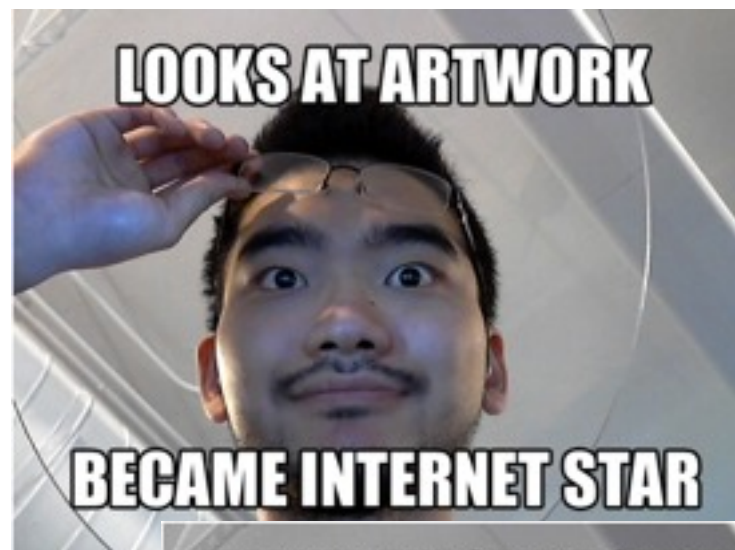
To help me know where you stand.



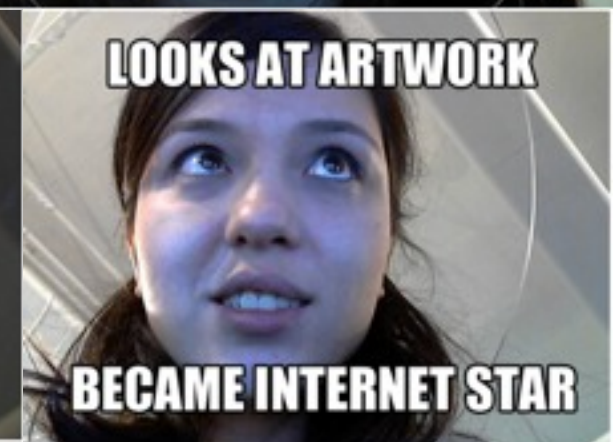
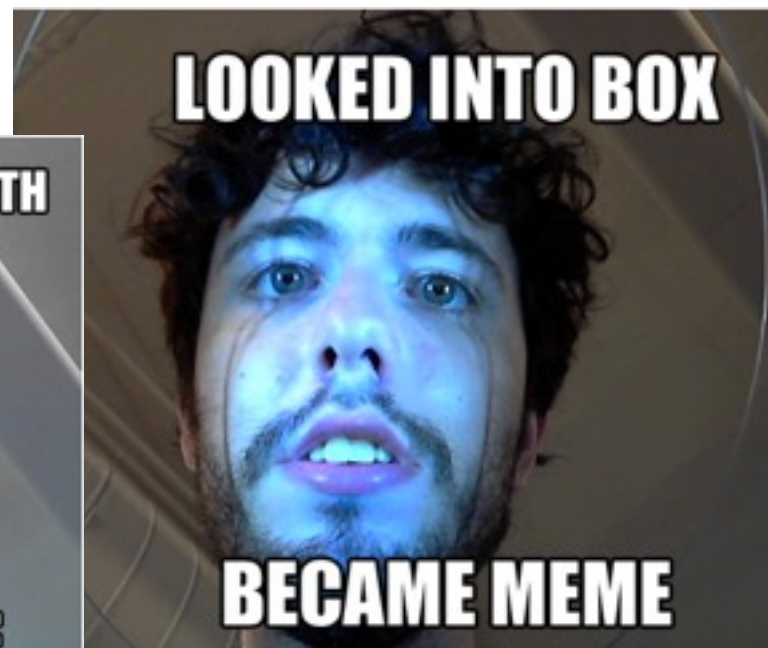
# assignment for next class

IT'S ABOUT TO GET REAL YALL

As you have probably already noticed, we love our memes.  
We've even had thesis projects that generate memes



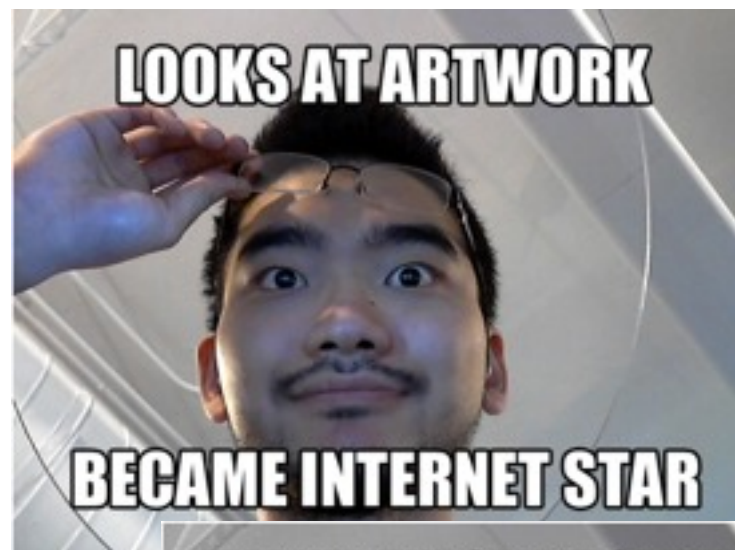
The Internet:  
It's Serious Business  
Brian Putz



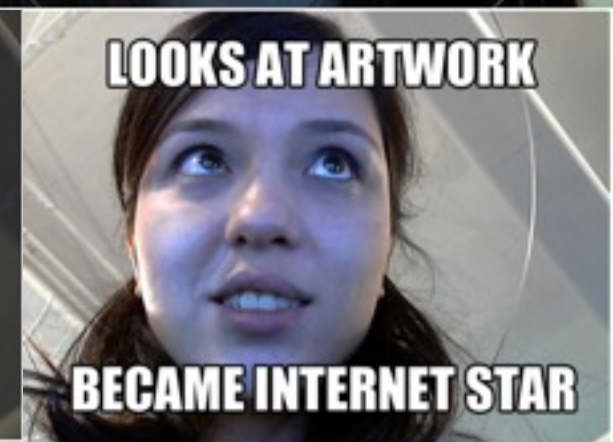
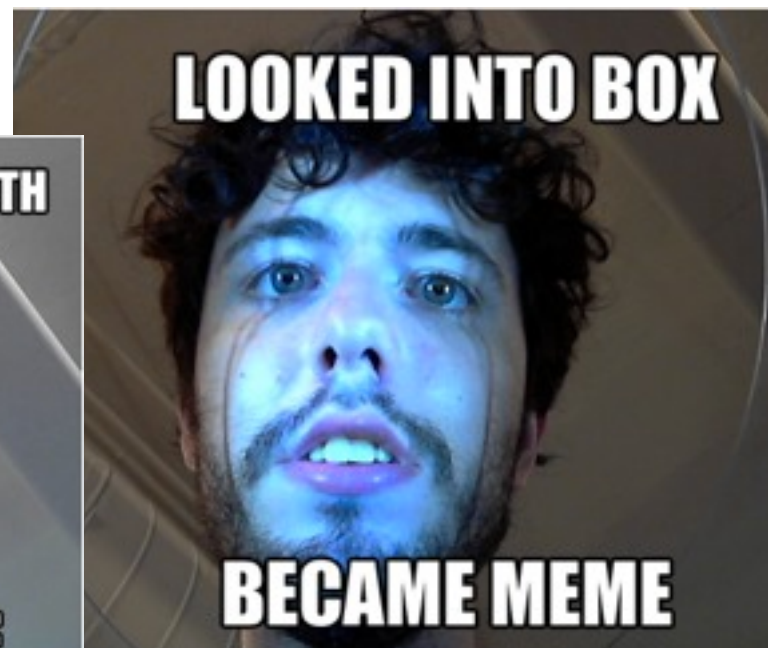
# assignment for next class

IT'S ABOUT TO GET REAL YALL

As you have probably already noticed, we love our memes.  
We've even had thesis projects that generate memes



The Internet:  
It's Serious Business  
Brian Putz

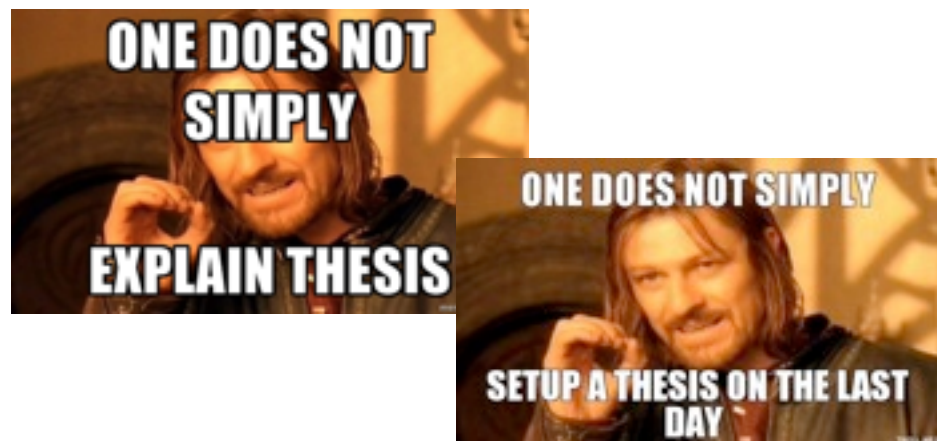




# assignment for next class

IT'S ABOUT TO GET REAL YALL

## Battle of the DT memes !



VS.



For your first assignment, create a battle of the memes. This does not have to be a game per se, though if it is, be sure to include a win state. You should feel completely free (perhaps even compelled) to make your own DT memes.

# assignment for next class

IT'S ABOUT TO GET REAL YALL

## Battle of the DT memes !

You must use all of the following in your code:

1. Arrays + for loops (you must access your array using a for loop)
2. PImage
3. A function you created
4. Motion (physics + gravity)
  - $\text{location} = \text{location} + \text{speed}$
  - $\text{speed} = \text{speed} + \text{gravity}$
  - \*\*Extra points for collision detection
5. Interactivity (whether mousePressed, keyPressed, etc)
6. You should also add a title to your battle somewhere in your sketch.

Upload your sketch to our classroom on **OpenProcessing**:

<http://www.openprocessing.org/classroom/1892>