

## Outline

- Processing
- Cool media art pieces
- Project proposals

Prof. Angela Chang

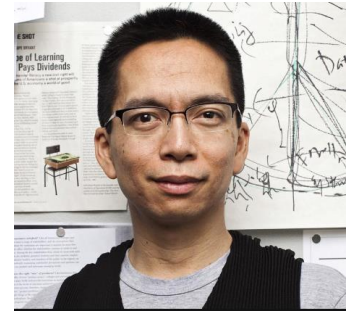
Lecture 16: Processing

Fall 2017. Nov 1

# CODE, CULTURE, AND PRACTICE

# Processing

IDE- Integrated Development Environment



1. IDE - Like a text editor for code, but allows more control such as running and stopping programs.
2. Designed for arts and design
3. Came out of Design By Number (John Maeda)
4. Written by Maeda's students Ben Fry and Casey Reas

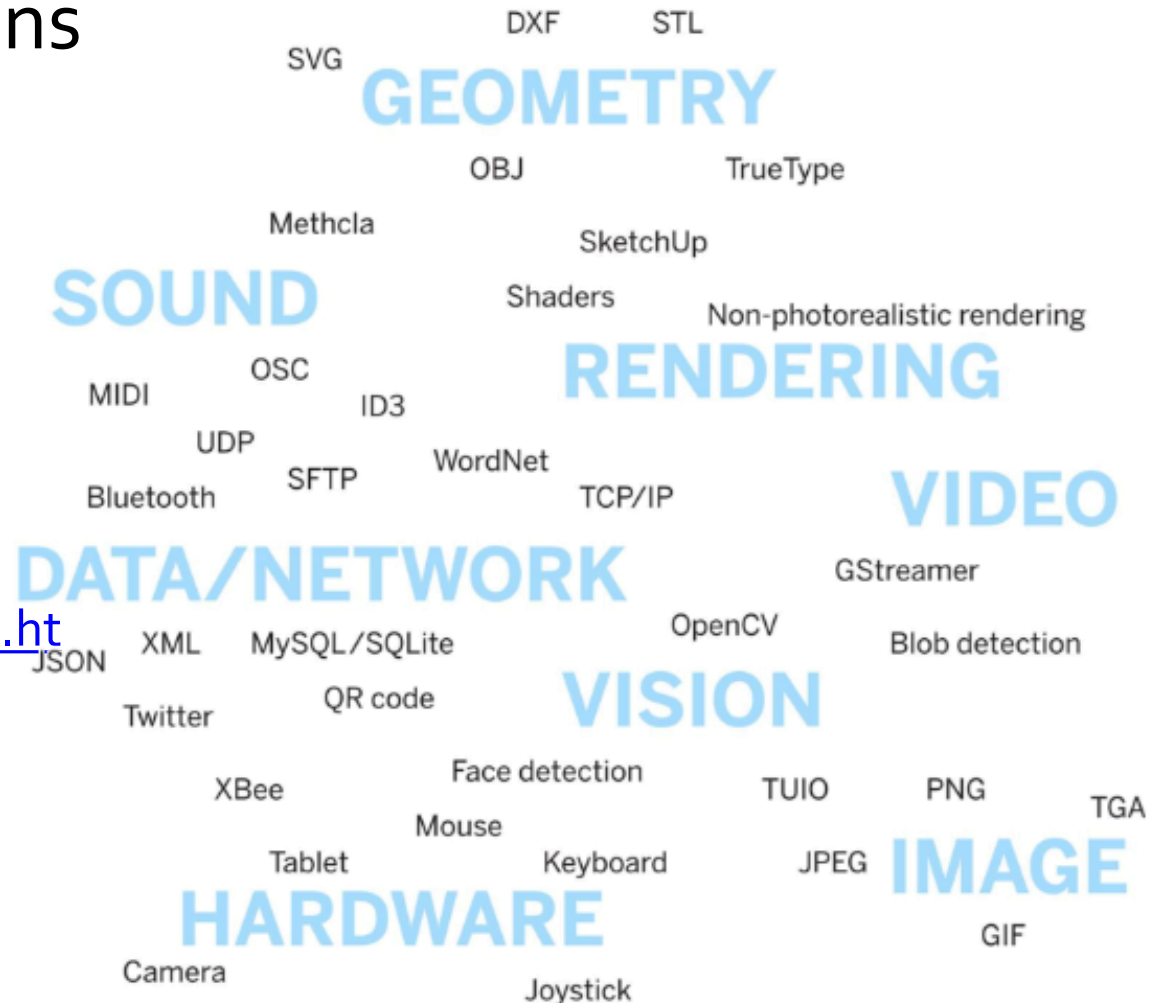


# Processing handles multimedia

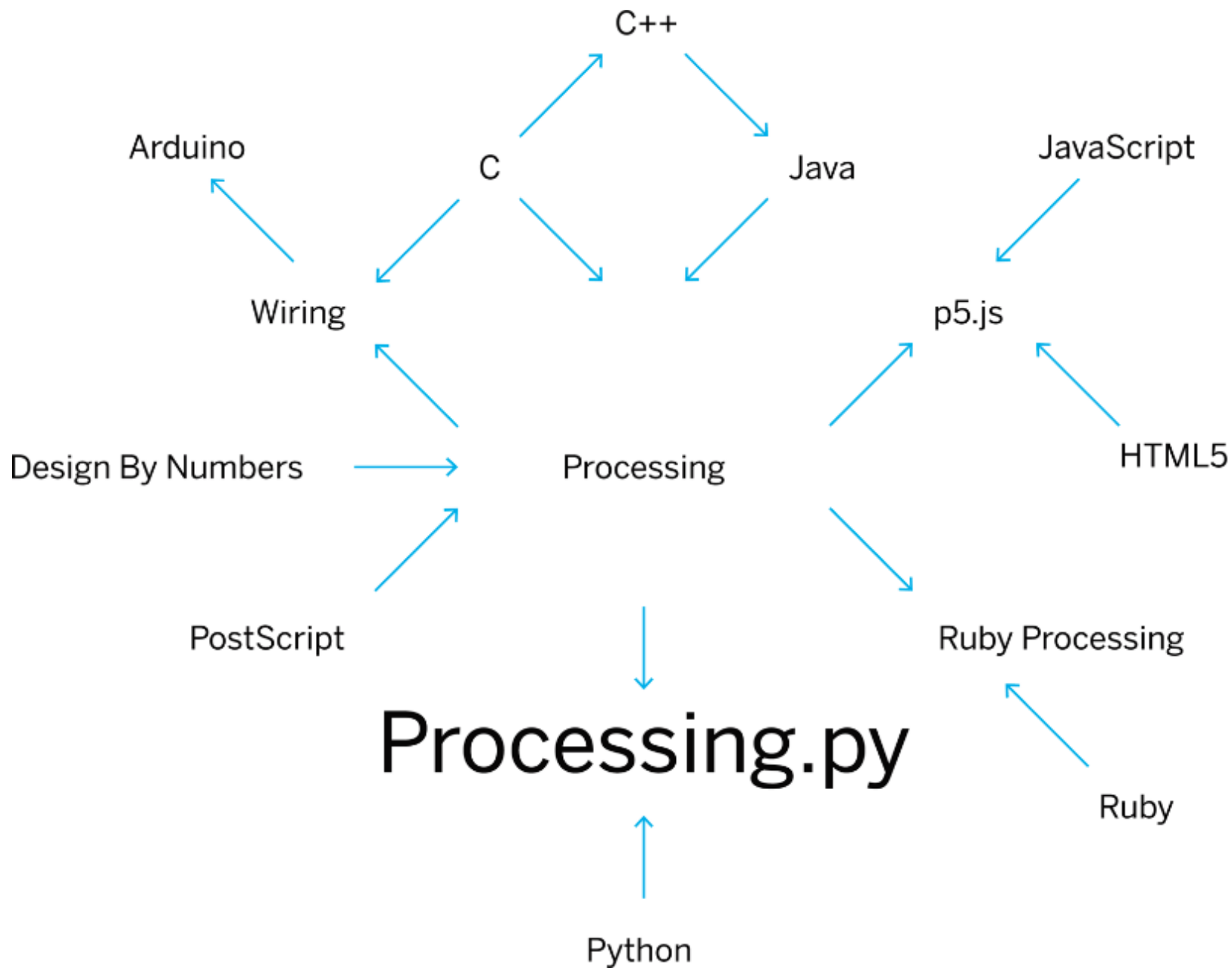
Open access to tons  
of libraries with  
example code:

For example, the  
Twitter Library

<http://twitter4j.org/en/index.html>



Most written in javascript... but some have been ported to python



# Using Python in the Processing IDE

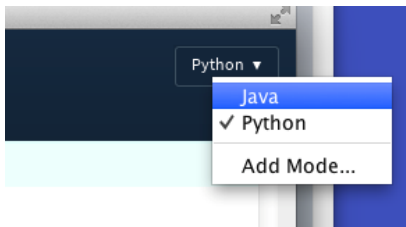
We're using Python mode  
inside the Processing IDE

<http://py.processing.org/>

Tutorials

<http://py.processing.org/tutorials/>

Tools → AddTool → “Add Mode”  
or  
click “Add Mode” on the top right of  
the sketch window to get Python mode.



Processing.py Tutorials. A collection of step-by-step lessons introducing Processing (with Python).

Many of these tutorials were directly translated into Python from their [Java counterparts](#) by the Processing.py documentation team and are accordingly credited to their original authors. [Please report any mistakes or inaccuracies in the Processing.py documentation GitHub.](#)



**Getting Started**  
by Casey Reas and Ben Fry

Welcome to Processing! This tutorial covers the basics of writing Python code.

Level: Beginner



**Processing Overview**  
by Ben Fry and Casey Reas

A more detailed introduction to the different features of Processing.

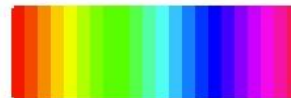
Level: Beginner



**Coordinate System and Shapes**  
by Daniel Shiffman

Drawing simple shapes and using the coordinate system.

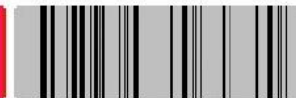
Level: Beginner



**Color**  
by Daniel Shiffman

An introduction to digital color.

Level: Beginner



**Interactivity**  
by Casey Reas and Ben Fry

Introduction to Interactivity with the Mouse and Keyboard

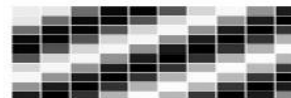
Level: Beginner



**Objects**  
by Daniel Shiffman

The basics of object-oriented programming.

Level: Beginner



**Two-Dimensional Lists**  
by Dan Shiffman

How to store data in a matrix using a two-dimensional list.

Level: Intermediate



**Images and Pixels**  
by Daniel Shiffman

How to load and display images as well as access their pixels.

Level: Intermediate



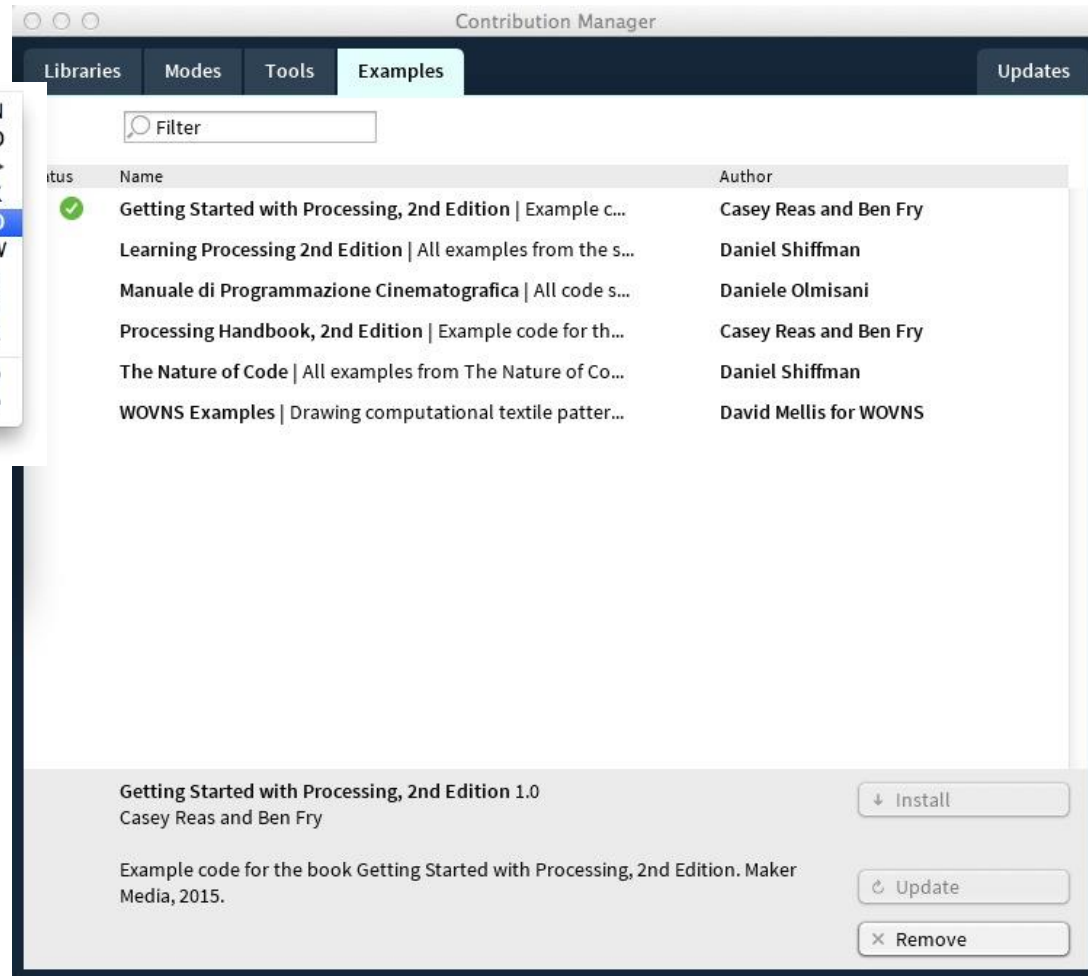
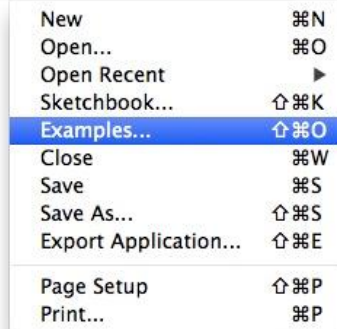
**Strings and Drawing Text**  
by Daniel Shiffman

Learn how to use the string class and display text onscreen.

Level: Intermediate

Why move to processing? More easily supports interactive demos.

lots of examples



# Python vs. Processing

1. code window

```
def mean(sequence):  
    sum = 0.0  
    for element in sequence:  
        sum = sum + element  
    return sum/len(sequence)
```

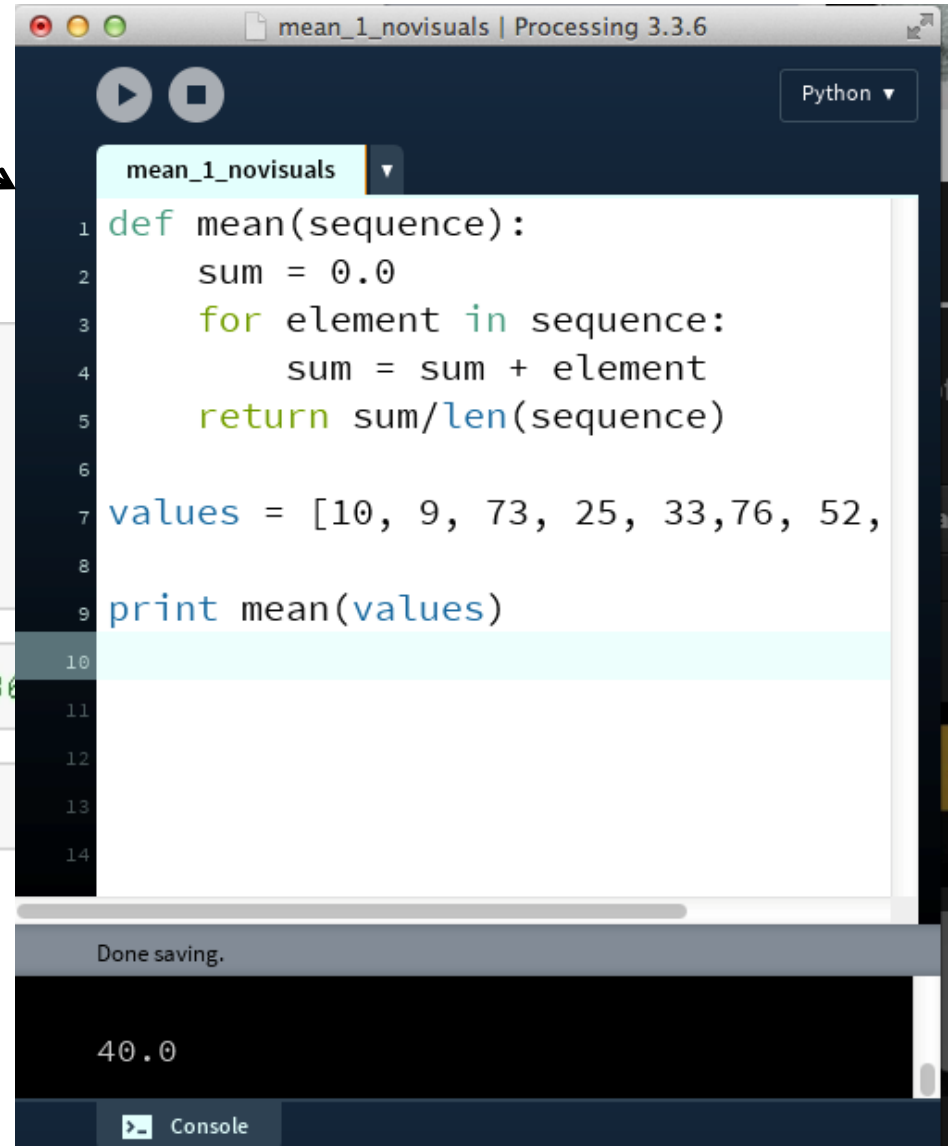
```
values = [10, 9, 73, 25, 33, 76, 52, 1, 35, 86]
```

```
print mean(values)
```

40.0

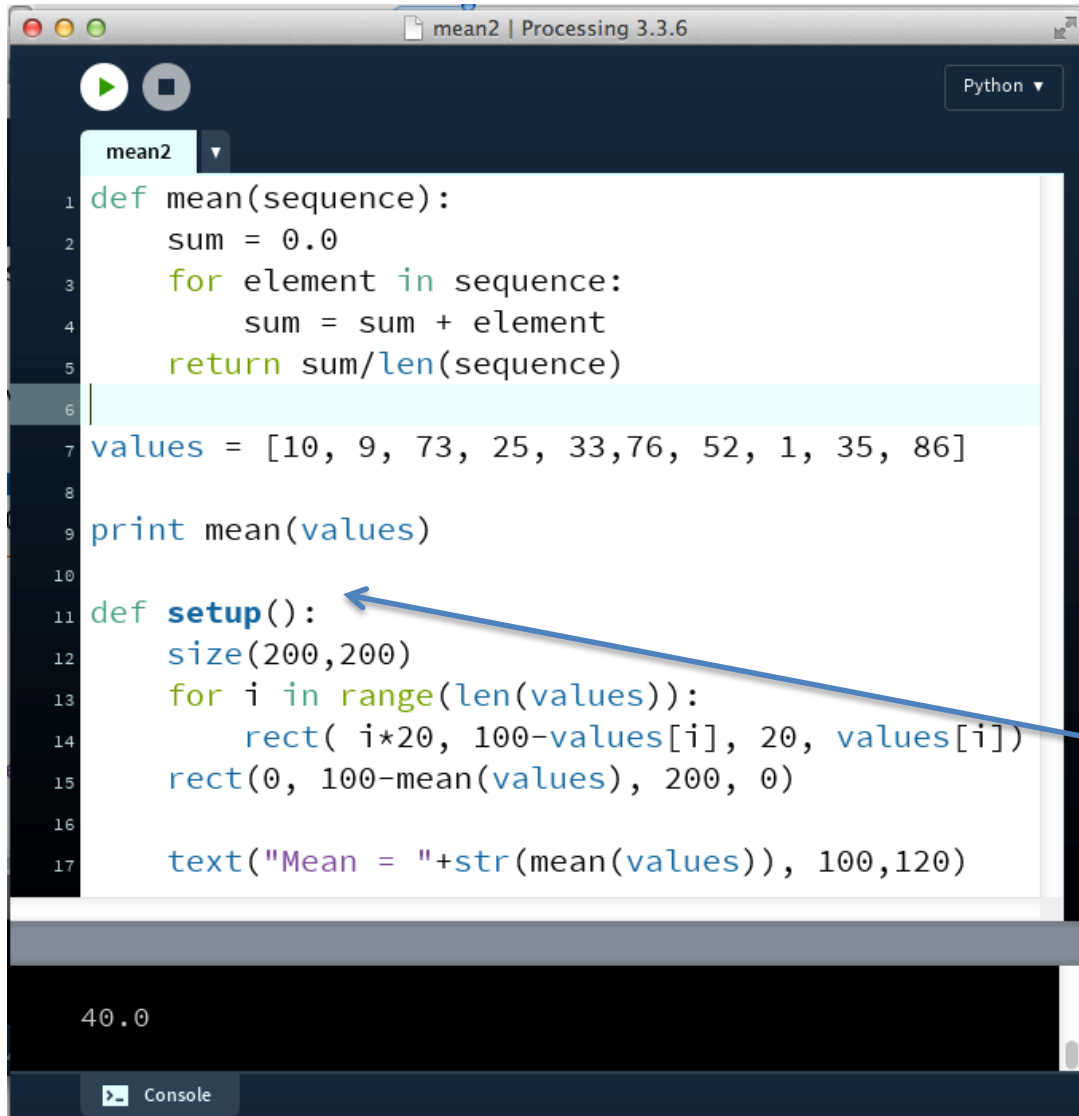
2. console output

computation result & errors



mean1/mean1.pyde

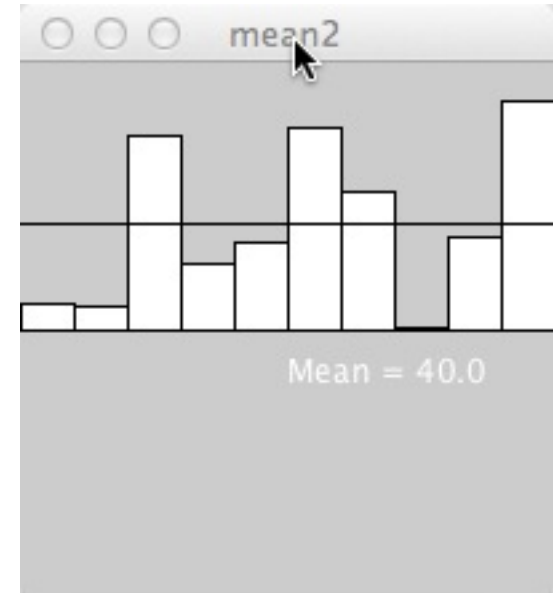
# The display window



```
mean2
1 def mean(sequence):
2     sum = 0.0
3     for element in sequence:
4         sum = sum + element
5     return sum/len(sequence)
6
7 values = [10, 9, 73, 25, 33, 76, 52, 1, 35, 86]
8
9 print mean(values)
10
11 def setup():
12     size(200,200)
13     for i in range(len(values)):
14         rect(i*20, 100-values[i], 20, values[i])
15         rect(0, 100-mean(values), 200, 0)
16
17     text("Mean = "+str(mean(values)), 100,120)
```

40.0

Console



3. display window

visual output  
explicitly coded

mean2/mean2.pyde



# Processing “sketches” have 2 new functions

for controlling the display window

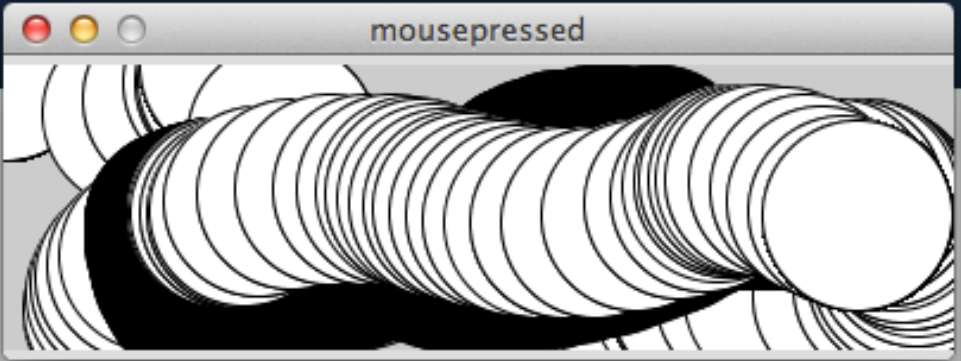
code window

mousepressed | Processing 3.3.6

display window

**setup()** sets up window or canvas

```
1 def setup():
2     size(400, 120)
3
4 def draw():
5     if (mousePressed):
6         fill(0)
7     else:
8         fill(255)
9     ellipse(mouseX, mouseY, 80, 80)
10
```



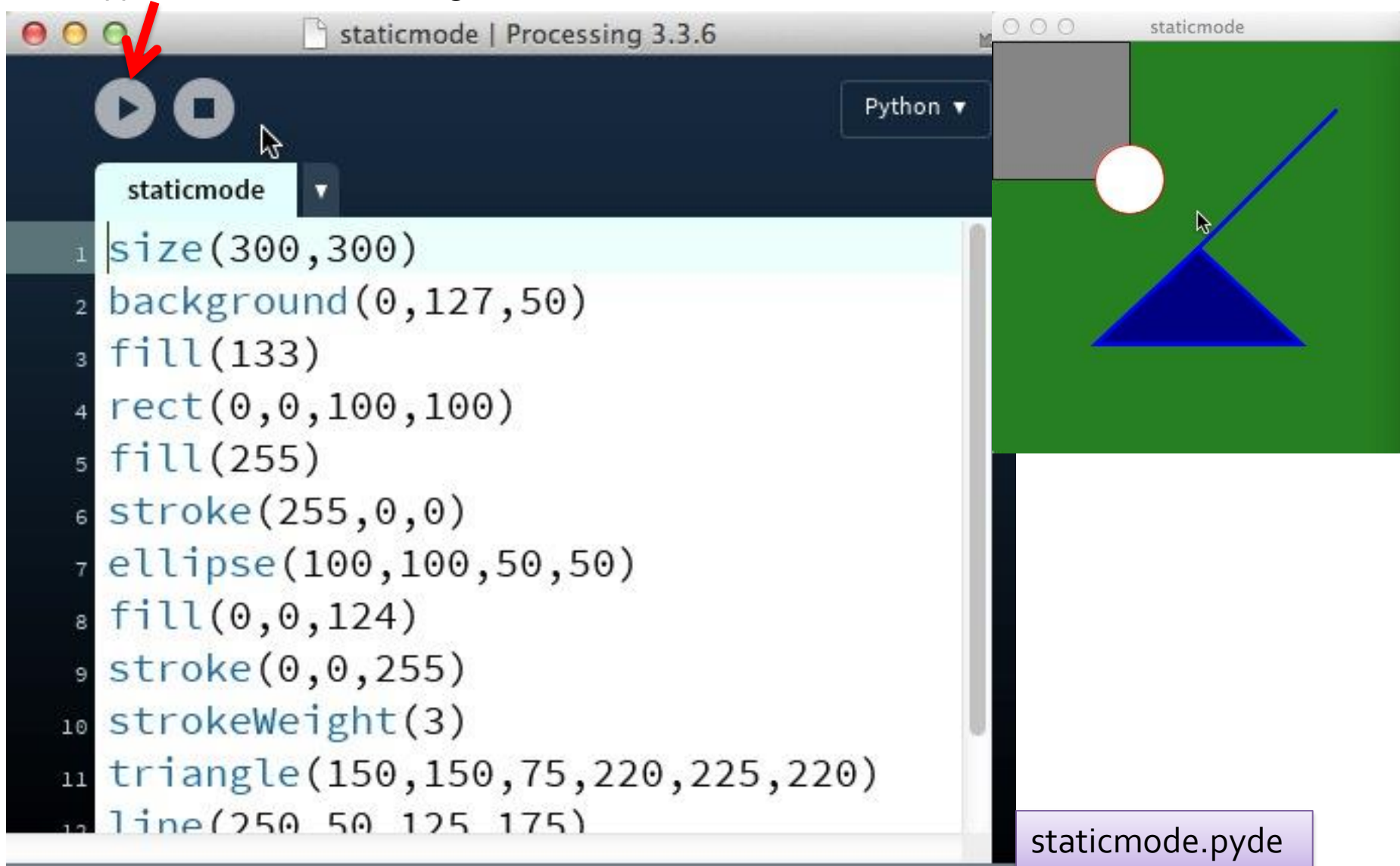
**draw()** updates display continuously (refreshes at 60Hz)

“If the mouse is pressed, draw a black circle.  
otherwise, draw a white circle where the mouse is”

mousepressed/mousepressed.pyde

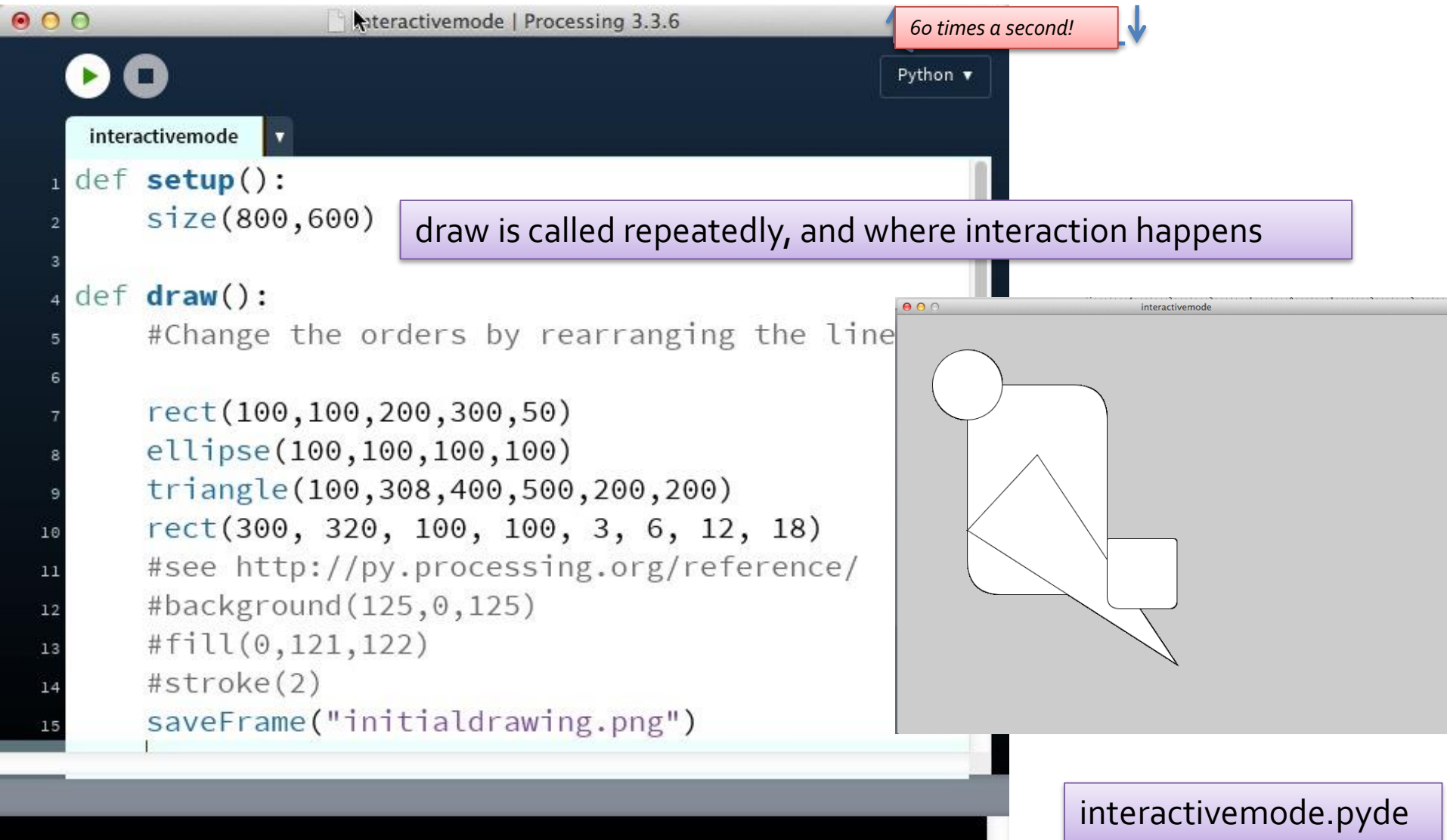
# static mode

Type code for 2D drawing in white area and click **Run** button.



# interactive mode

click run button → processing runs `setup()` → processing runs `draw()`



interactivemode | Processing 3.3.6

60 times a second!

Python ▾

interactivemode ▾

```
1 def setup():
2     size(800,600)
3
4 def draw():
5     #Change the orders by rearranging the line
6
7     rect(100,100,200,300,50)
8     ellipse(100,100,100,100)
9     triangle(100,308,400,500,200,200)
10    rect(300, 320, 100, 100, 3, 6, 12, 18)
11    #see http://py.processing.org/reference/
12    #background(125,0,125)
13    #fill(0,121,122)
14    #stroke(2)
15    saveFrame("initialdrawing.png")
```

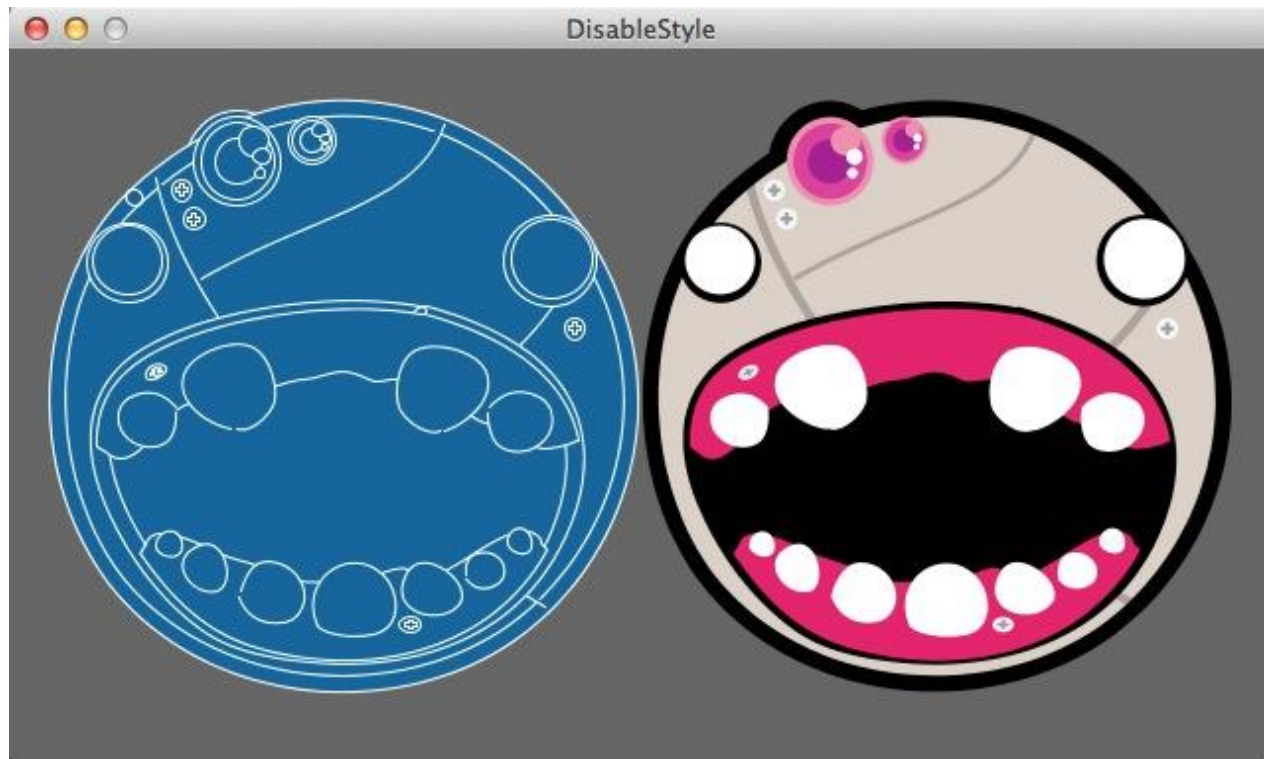
draw is called repeatedly, and where interaction happens

interactivemode

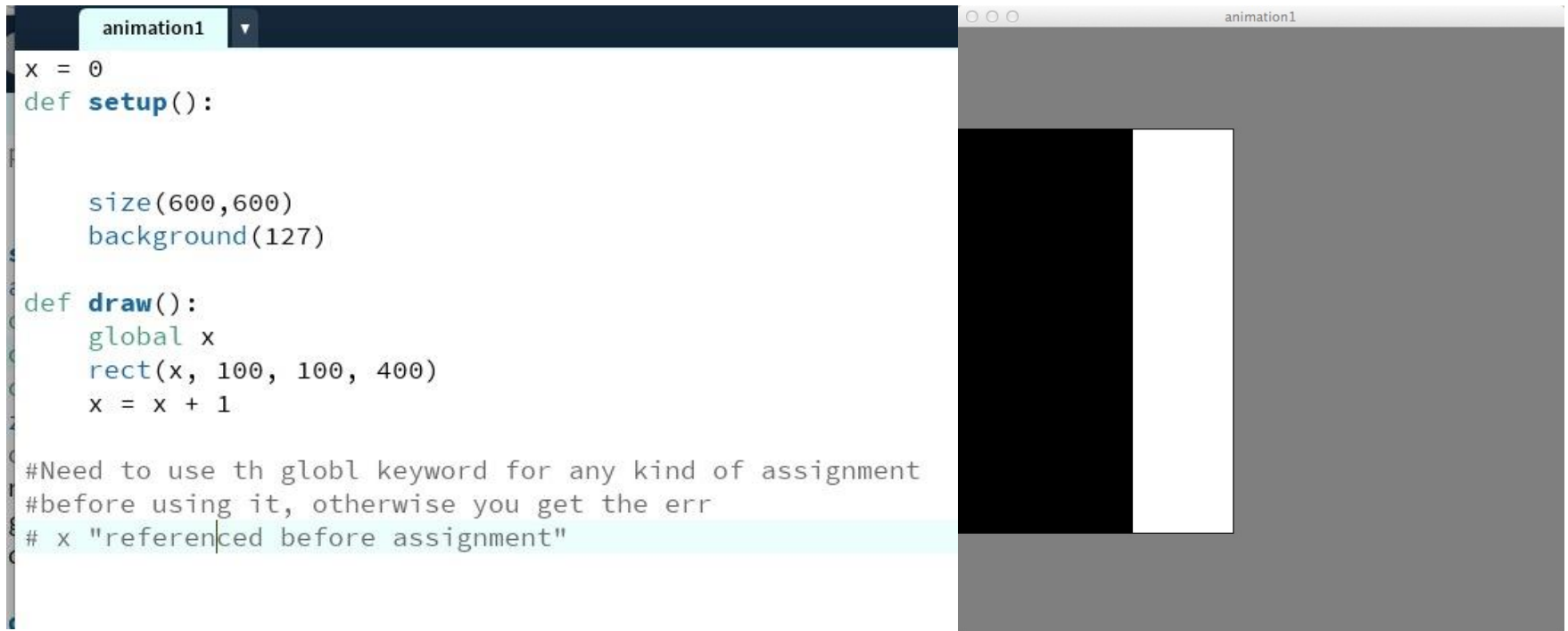
interactivemode.pyde

# Class participation today

- Play and make two 2d pictures with Processing
- Upload to canvas:
  - pyde source code
  - graphics you generated



# Animation- images that move



can you change the speed?

can you make it change motion direction?

can you make it bounce?

can you make it change color?

animation1.pyde

# Bounce

```
bounce ▼  
  
x = 0  
direction = 1  
def setup():  
    size(600,600)  
    background(127)  
  
def draw():  
    global x  
    global direction  
    rect(x, 100, 100, 400)  
    x = x + direction * 5  
    if x >= 480:  
        direction = -1  
    if x <= 0:  
        direction = 1
```

create a direction flag

at every frame, update the position by the direction

change direction when boundary has been reached

"when x goes beyond 480"  
-make the step direction negative

"when x goes beneath 0"  
-direction changes to positive

bounce.pyde



# HW for today

Write a creative program in processing to animate text and images.

Have your program take text files and images to produce an experience that has elements of each.

Play and make an animation with Processing

- Check out loadStrings
- Check out loadImage
- Use StringImage as a template to alter the image and text file
- Upload to canvas:
  - a zip file containing
  - pyde source code
  - text source
  - image source file

```
LoadStrings ▼
#http://py.processing.org/reference/loadStrings.html
lines = loadStrings("lovesong.txt")
print("there are %i lines" % len(lines))
for line in lines:
    print(line)

lines = loadStrings("http://processing.org/about/index.html")
print("there are %i lines" % len(lines))
for line in lines:
    print(line)
```

```
LoadImage ▼
#http://py.processing.org/tutorials/pixels/
def setup():
    global img
    size(320,240)
    # Make a new instance of a PImage by loading an image
    # Declaring a variable of type PImage
    img = loadImage("lunar.jpg")

def draw():
    global img
    background(0);
    # Draw the image to the screen at coordinate (0,0)
    image(img,0,0)
```

LoadImage.pyde  
Loadstrings.pyde

StringImage.pyde

due on Monday

# Homework

1

Play and make two 2d pictures with Processing, upload your source to canvas

<http://bit.ly/2017Lecture18Processing2D>

Alter StringsImage to use your own mashup animation of images and strings.

<http://bit.ly/2017HW8TextImageMashup>

due Monday

2

Take a look at the presentations and feedback from today's class. Start thinking about the steps for pursuing each project, finding information and resources in regards to the proposed projects. Create a repository of information and organize it together. Decide whether you do solo project together or alone later. If you haven't already done so, fill out the group participation homework & peer review survey

<http://bit.ly/2017FirstProjectProposal>





# Homework: Research your topic

We'll form into groups to pursue the projects towards some form of realization

- I'll send out email with groups by this weekend.
- Reflect on the feedback and messages presented in class.
- As a group, look online and create media documenting effective messages, media works, and resources relevant to those heard in class. This can be an abstract, collage, and/or file repository.
- Post your collaborative research in an online repository of some sort (**Wordpress blog, Dropbox, or Github**). Upload a summary document describing the resources you've found, any analysis you've done, and the link to the resources.
- Share your group findings at end of next class on Wednesday 11/8.



# Summary of today

- Technical stuff- Processing IDE
  - Integrated Development Environments (contains an output area)
  - How to install it on your machine
- Your first works in Processing
  - Class participation credit for today – 2D images  
<http://bit.ly/2017Lecture18Processing2D>
  - HW making a text & image mashup  
<http://bit.ly/2017HW8TextImageMashup>
- Group work pending– forming groups
  - fill out that peer survey, and I'll form the groups by this Friday
  - <http://bit.ly/2017FirstProjectProposal>
  - Start sharing information about project directions
  - Groups start to create and share your media repository