Lecture 27:

ANIMATION

CSC111: Introduction to CS through Programming

R. Jordan Crouser

Assistant Professor of Computer Science

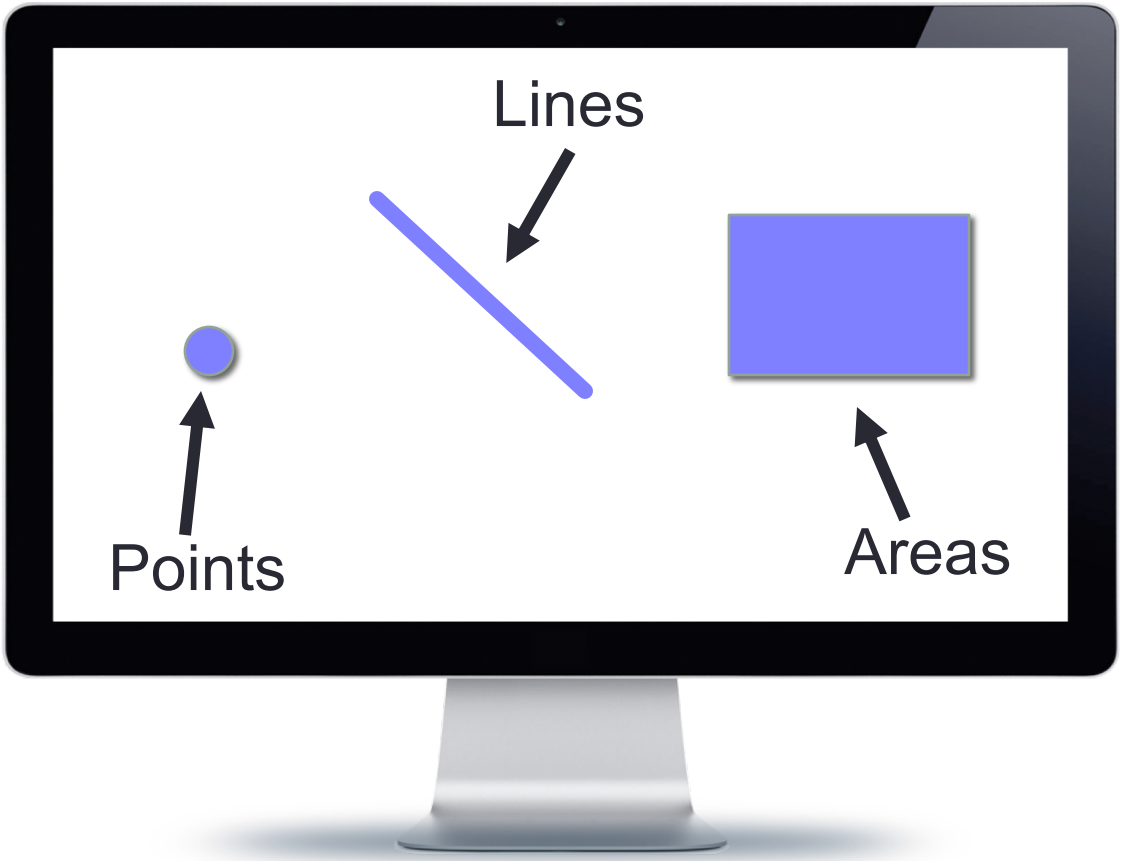
Smith College

Outline

- ✓Python packages (graphics)
- Animation basics
  - understanding motion
  - the `.move()` method
  - Keeping objects on the screen
- Lab: Fish Tank
- Interaction

✓ Draw stuff

"graphical primitives"

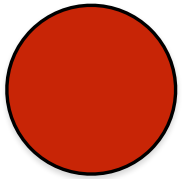


✓ Draw stuff

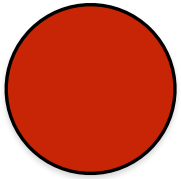
using the **`graphics`** module

2. Make it move

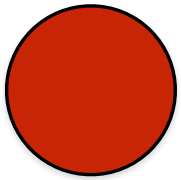
Animation basics



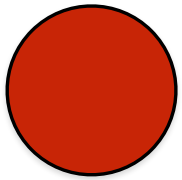
Animation basics



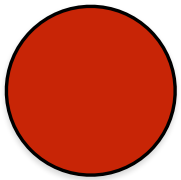
Animation basics



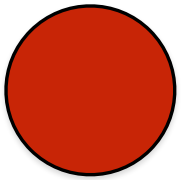
Animation basics



Animation basics



Animation basics

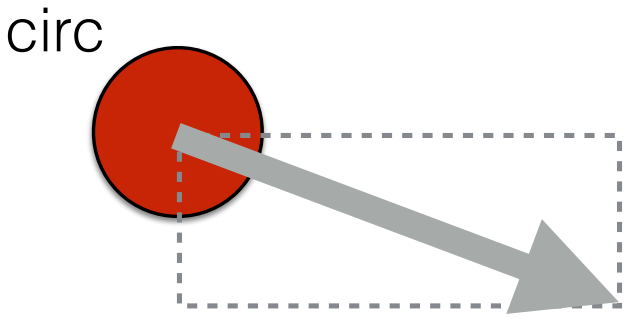


Discussion

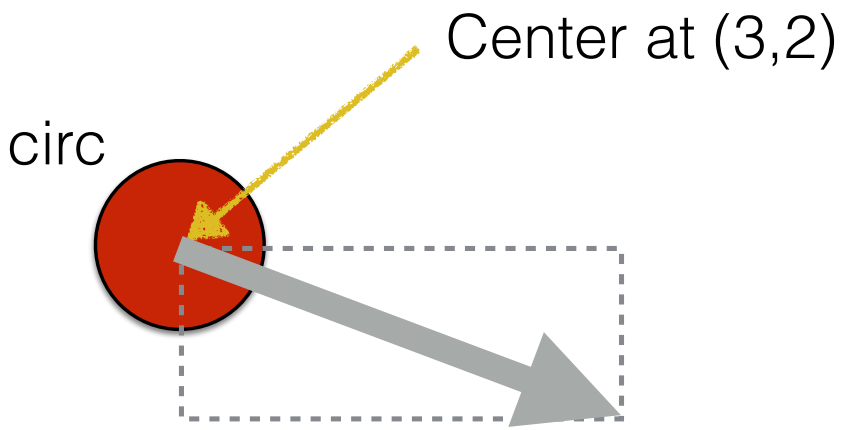
What do I need to **be able do**

to make that happen?

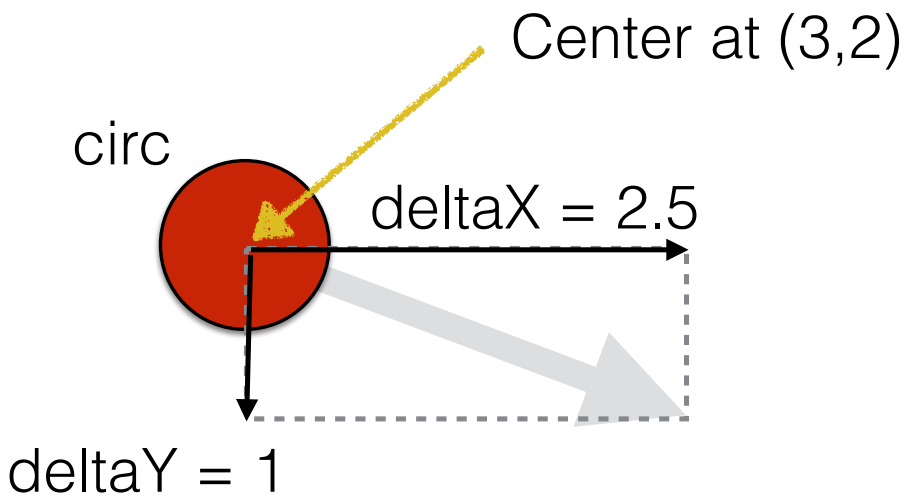
Understanding motion



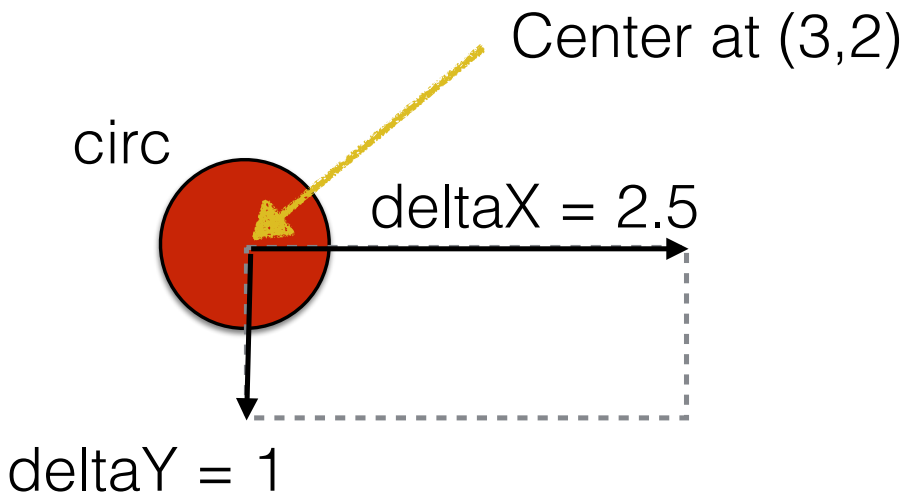
Understanding motion



Understanding motion

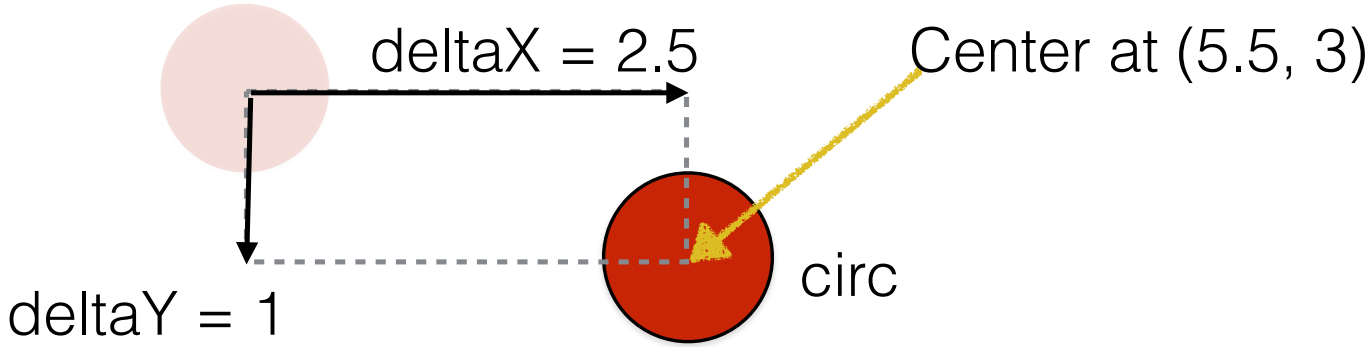


The `.move()` method



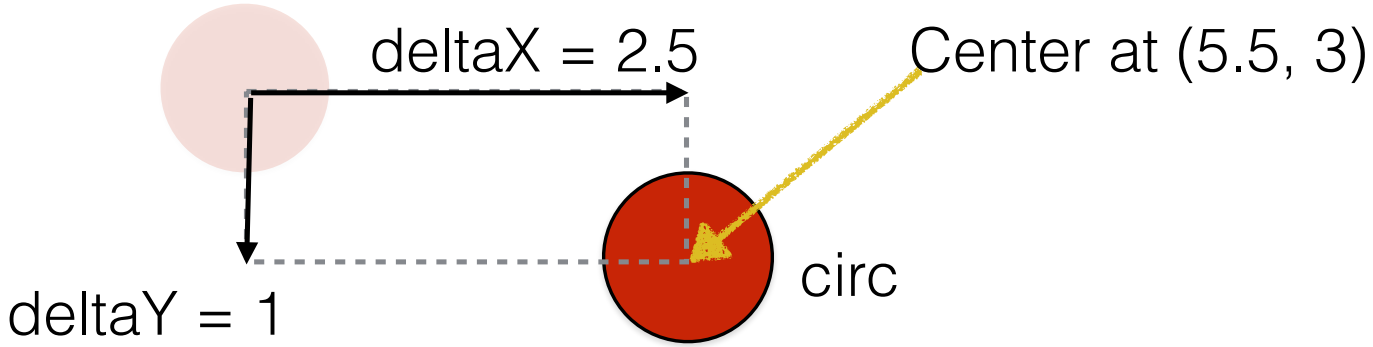

```
circ.move( deltaX, deltaY )
```

The `.move()` method



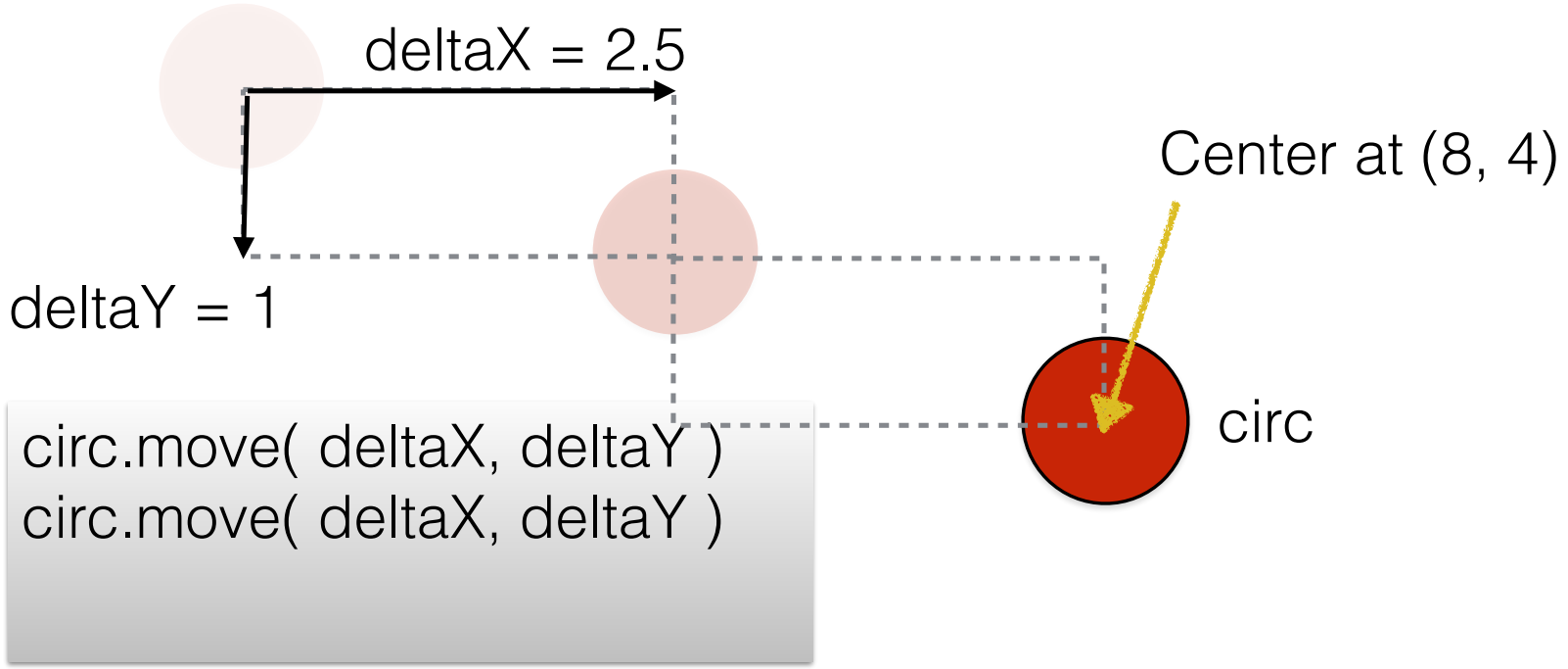

```
circ.move( deltaX, deltaY )
```

Animation: call `.move()` method >1x




```
circ.move( deltaX, deltaY )
circ.move( deltaX, deltaY )
```

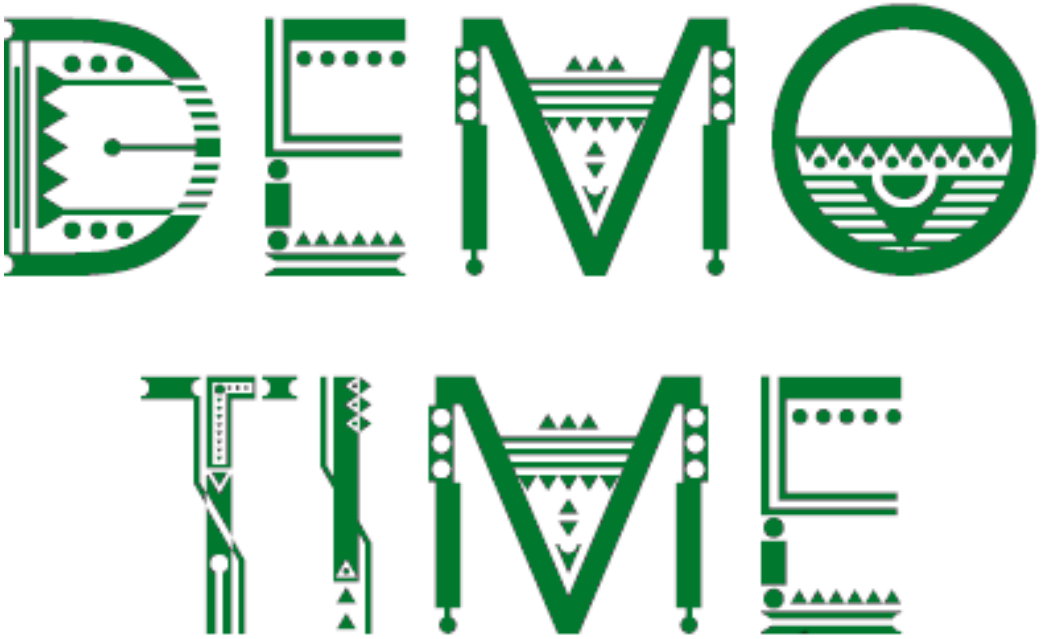
Animation: call `.move()` method >1x



Basic organization of animation `main()`

```
def main():
    # 1. open the graphics window
    # 2. define/initialize graphic objects
    # 3. start animation loop, stop on
    #    specific user interaction
    while win.checkMouse() == None:
        # 4. move/update each object
    # Loop is over.
    # 5. close the graphic window
```

Our first animated **`graphics`** program



Discussion

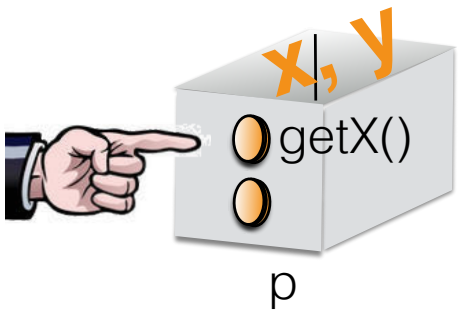
How do we keep an object from

moving off the screen?

Every **`graphics`** element is an **`Object`**…

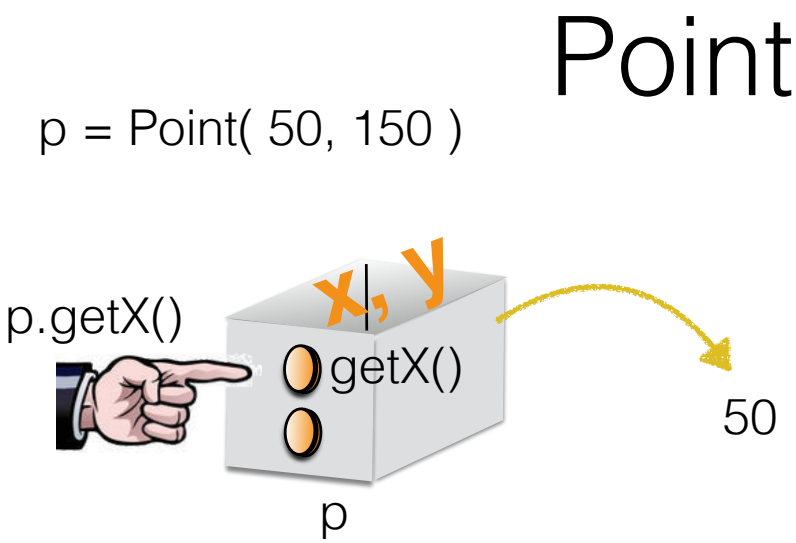
Point

p = Point(50, 150)



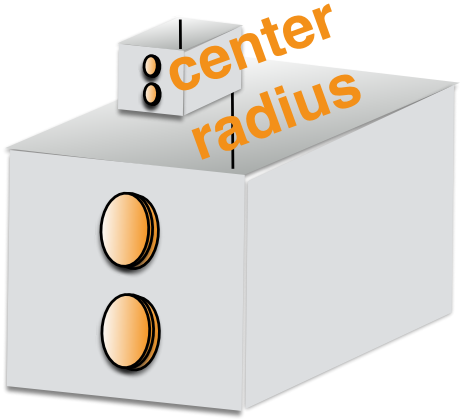
p

Every **`graphics`** element is an **`Object`**…



Every **`graphics`** element is an **`Object`**…

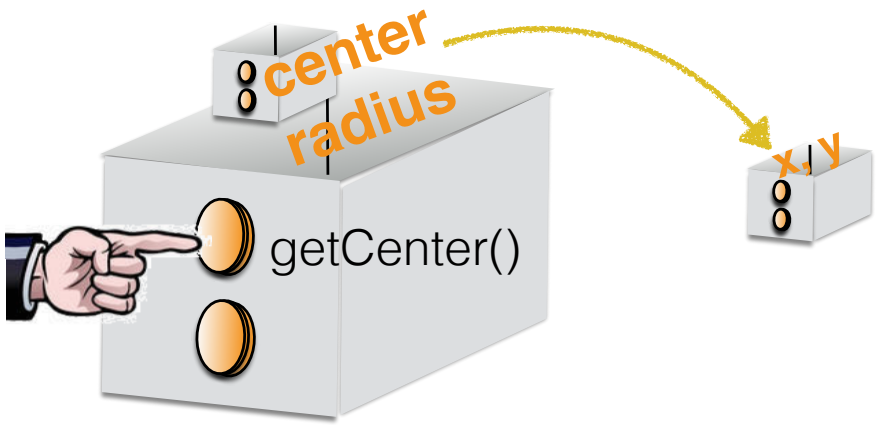
Circle



Circle Object

Every **`graphics`** element is an **`Object`**…

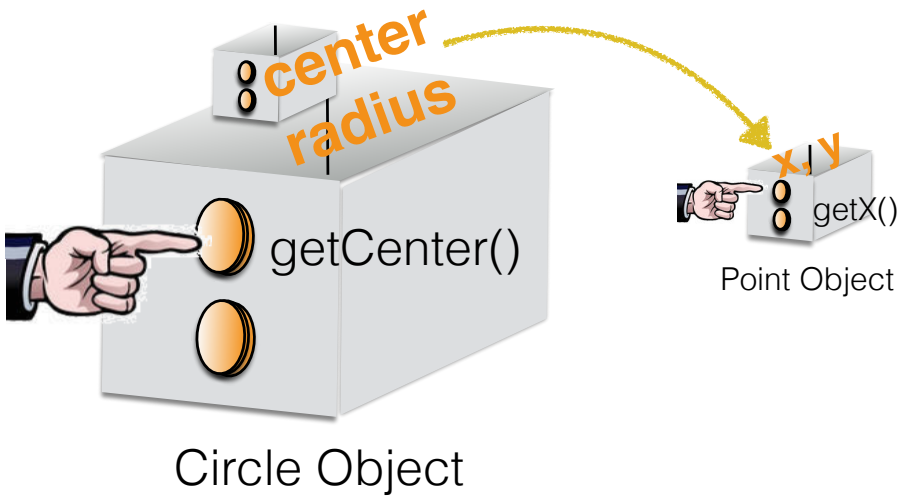
Circle



Circle Object

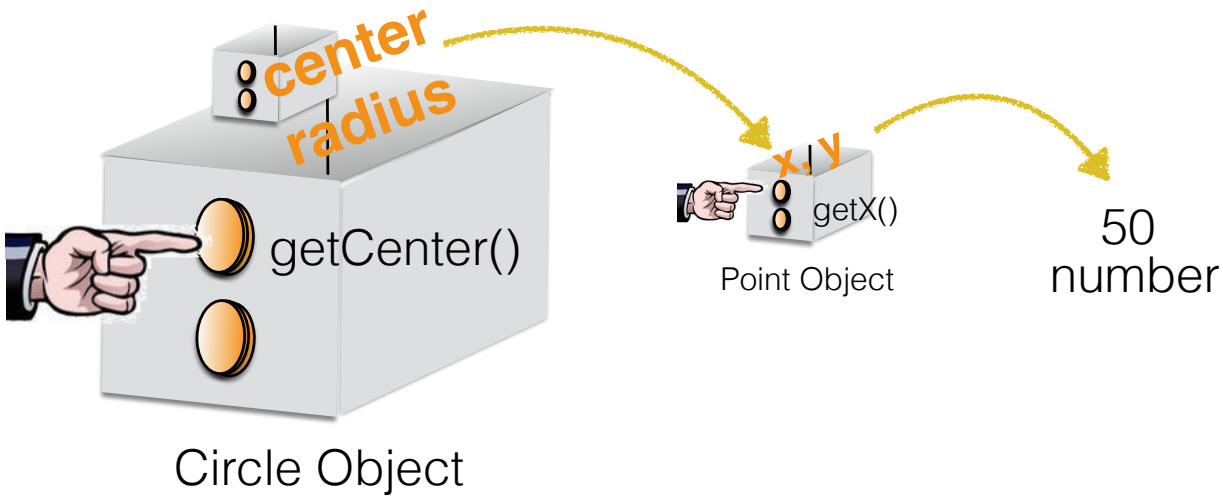
Every **`graphics`** element is an **`Object`**…

Circle



Every **`graphics`** element is an **`Object`**…

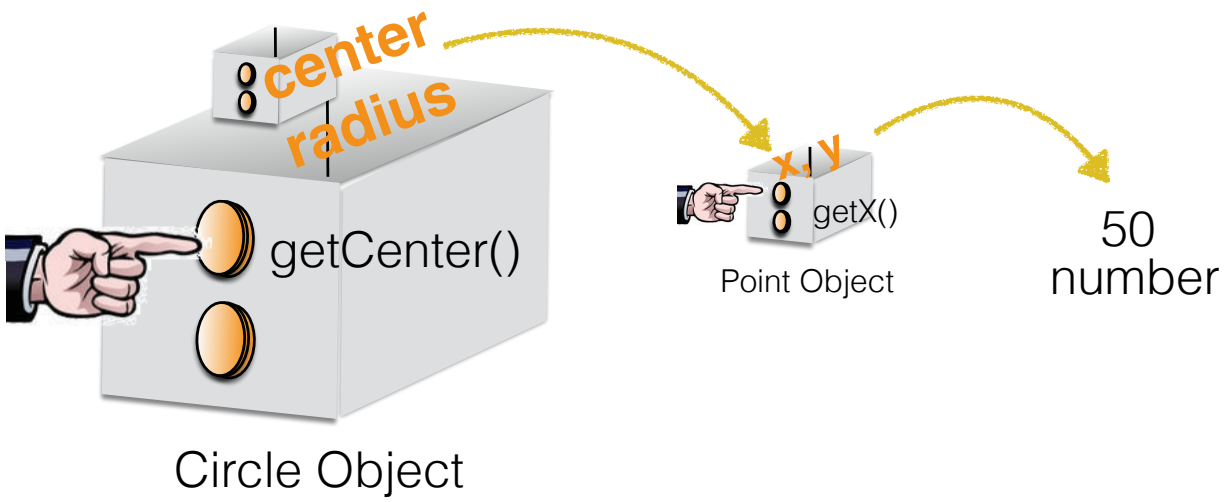
Circle



Every **`graphics`** element is an **`Object`**…

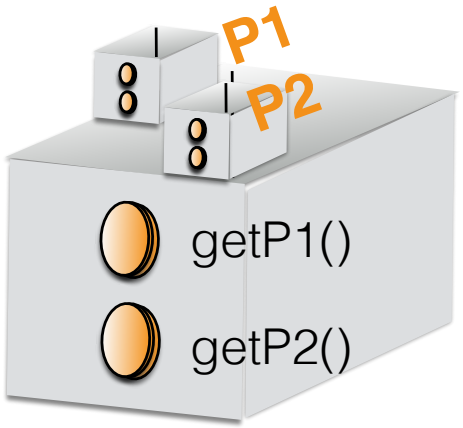
Circle

```
x = circ.getCenter().getX()
```



Every **`graphics`** element is an **`Object`**…

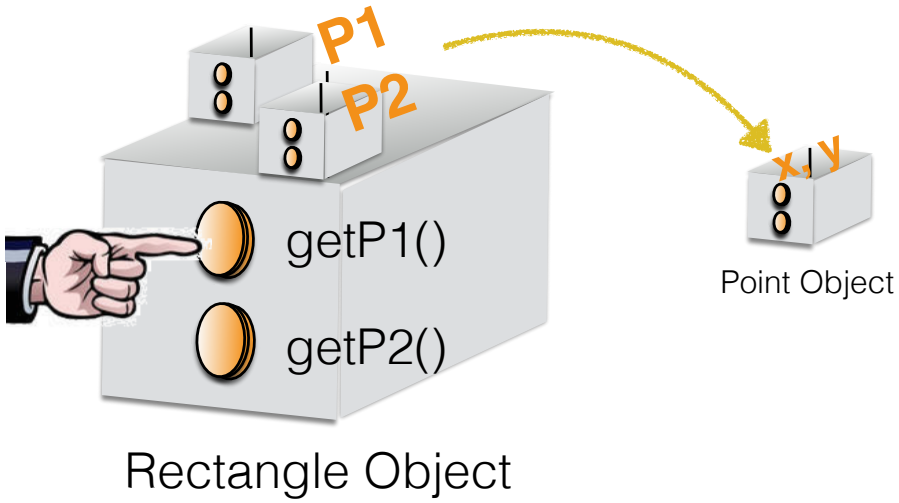
Rectangle



Rectangle Object

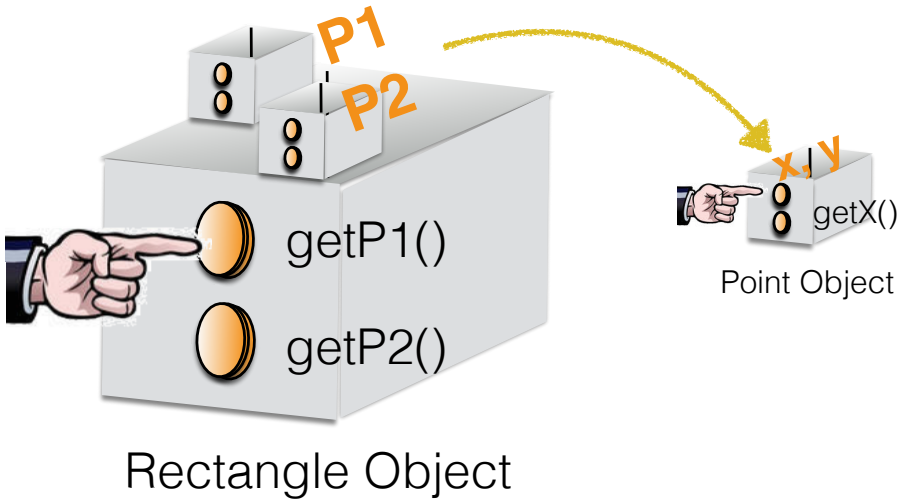
Every **`graphics`** element is an **`Object`**…

Rectangle



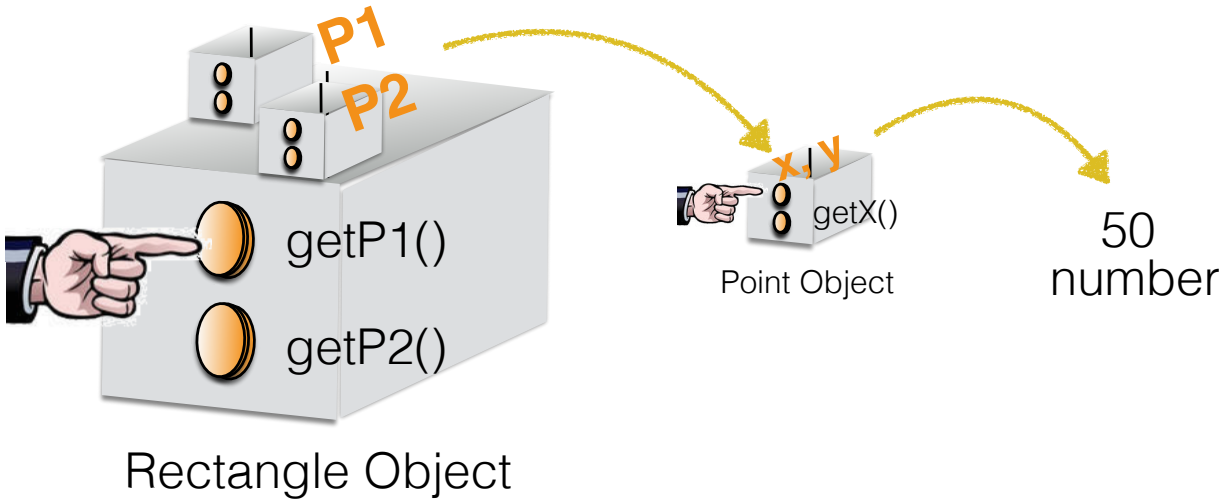
Every **`graphics`** element is an **`Object`**…

Rectangle



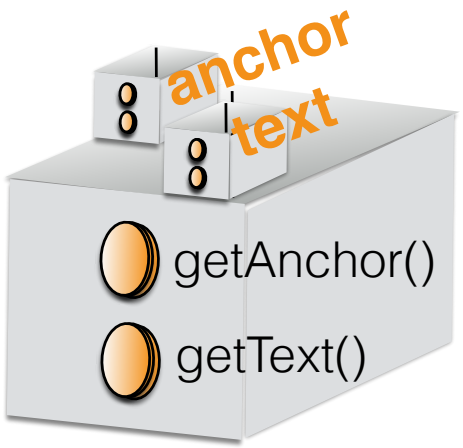
Every **`graphics`** element is an **`Object`**…

Rectangle



Every **`graphics`** element is an **`Object`**…

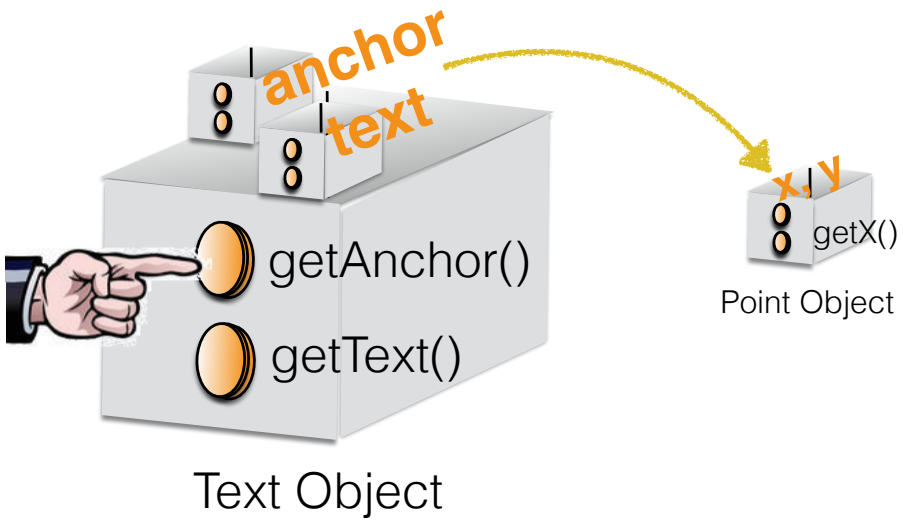
Text (label)



Text Object

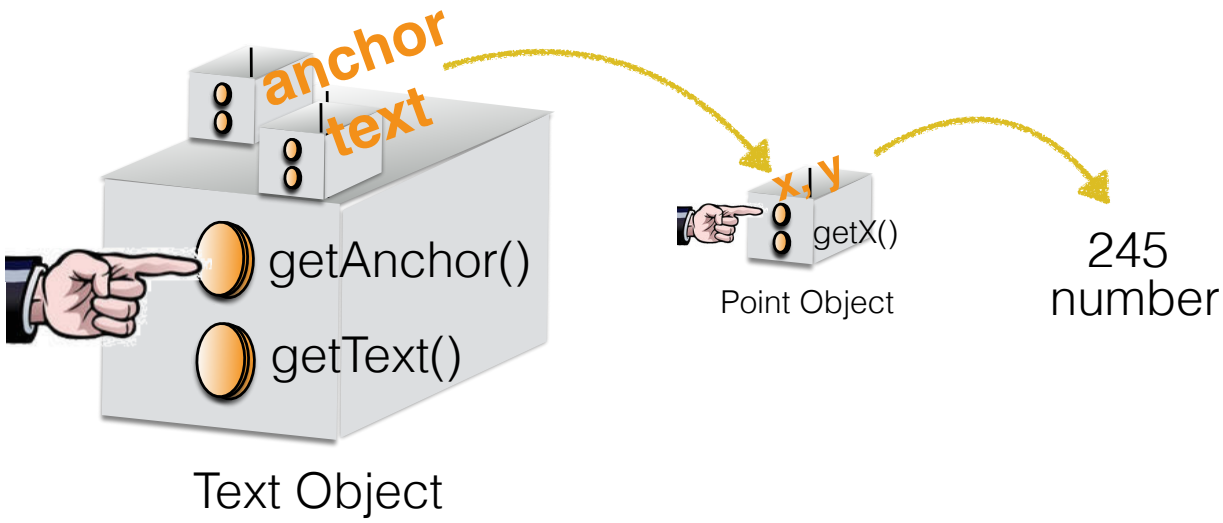
Every **`graphics`** element is an **`Object`**…

Text (label)



Every **`graphics`** element is an **`Object`**…

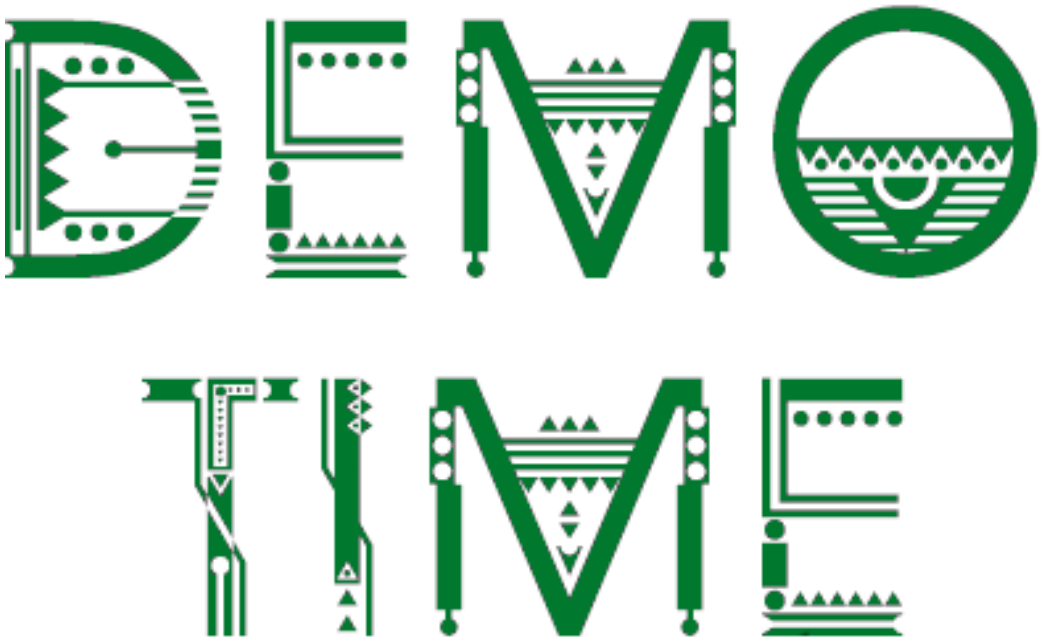
Text (label)



Discussion (again)

Using this, how do we keep an object from **moving off the screen**?

Bouncing off the walls



Outline

- ✓Python packages (graphics)
- ✓Animation basics
  - ✓understanding motion
  - ✓the `.move()` method
  - ✓Keeping objects on the screen
- Lab: Fish Tank
- Interaction

Lab 9: Fish Tank

Animation

Jordan

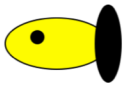
Secure https://jcrouser.github.io/SCS-Noonan-CSC/labs/lab-15-animation.html

For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...

- Refresher
- Classy graphics
- Challenge 1: Aquarium
- Animated objects
- Challenge 2: Swimming Fish
- Final Challenge
- Submission of Lab 15

Classy graphics

This is pretty cool, I guess... but would be way more fun to animate a more complex object like a taxi or maybe a cute little fish:



This little guy is made up of just two ovals and a circle. However, it doesn't seem quite right to draw each object independently. After all, they're all part of the same `object`. And what if we wanted to move it around or draw more than one? This calls for a `class`!

Let's make a simple `Fish` class, containing just `__init__` and `draw` methods:

```
from graphics import *
import random
class Fish:
  """Definition for a fish with a body, eye, and tail"""
  def __init__(self, win, position):
    """constructs a fish made of 1 oval centered at `position`,
    a second oval for the tail, and a circle for the eye"""

    red = random.randint(0,255)
    green = random.randint(0,255)
    blue = random.randint(0,255)

    # body
    p1 = Point(position.getX()-40, position.getY()-20)
    p2 = Point(position.getX()+40, position.getY()+20)
    self.body = Oval( p1, p2 )
    self.body.setFill( "yellow" )

    # tail
    p1 = Point(position.getX()+30, position.getY()-30)
```