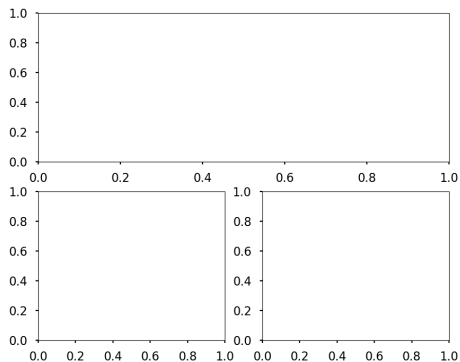


# Announcements

- ▶ Homework
  - ▶ Homework 10 posted! (Actually this time!)
  - ▶ I'm still working on getting HW9 graded
- ▶ Project info coming Friday
  - ▶ Poll will be going out to you to get some feedback on potential groups
- ▶ Polling: `rembold-class.ddns.net`

# Review Question

Which block of code to the right will produce the arrangement of axes below?



A) `ax1 = f.add_subplot(222)`  
`ax2 = f.add_subplot(221)`  
`ax3 = f.add_subplot(212)`

B) `ax1 = f.add_subplot(223)`  
`ax2 = f.add_subplot(221)`  
`ax3 = f.add_subplot(122)`

C) `ax1 = f.add_subplot(224)`  
`ax2 = f.add_subplot(223)`  
`ax3 = f.add_subplot(211)`

D) `ax1 = f.add_subplot(213)`  
`ax2 = f.add_subplot(211)`  
`ax3 = f.add_subplot(222)`

# Grouping Image Components

- ▶ Drawings make an easy way to use classes to make more complicated objects!
- ▶ Break the process of creating the drawing into multiple methods
  - ▶ Maybe one to create the parts of the image
  - ▶ One to color the parts of the image
  - ▶ One to draw the parts of the image
- ▶ Will be very useful for later creating movement and animation
  - ▶ Can add motion, physics, etc to each object

# Adding Motion

- ▶ Can use `.move(dx,dy)` to move a graphical element by a  $dx$  and  $dy$
- ▶  $dx$  and  $dy$  are a displacement!
  - ▶ Shift from the current location
  - ▶ You can not directly tell it to move to a particular point
    - ▶ Would need to find the difference between where it is at and where you want it to go.

# Animation: Method 1

- ▶ Continuous motion achieved through repeated `.move` calls
- ▶ One method would be inside a loop
- ▶ You **need** some way to regulate the speed of the movement
  - ▶ The drawing will not be able to visibly keep up with Python otherwise
  - ▶ Can use `update()` in loops
    - ▶ Will force an update to the motion at that point
    - ▶ Can pass in a rate for the number of times it will update per second

# Animation: Method 2

- ▶ Often nice to bundle movement commands into a class, where it is cumbersome to call that method each time in the loop
- ▶ Might also want movement to still be happening while the program waits for some mouse or key press
- ▶ Can use a recursive call which tells the graphics window to call this same move method again after a specified delay.
  - ▶ Use `<window obj>.after(<delay>, <function>)`
  - ▶ The window will figure out the timing to move the requested objects when it has a spare moment