# Intro to Coding with Python– Interaction

Dr. Ab Mosca (they/them)

Slides based off slides courtesy of Jordan Crouser (https://jcrouser.github.io/)

# Plan for Today

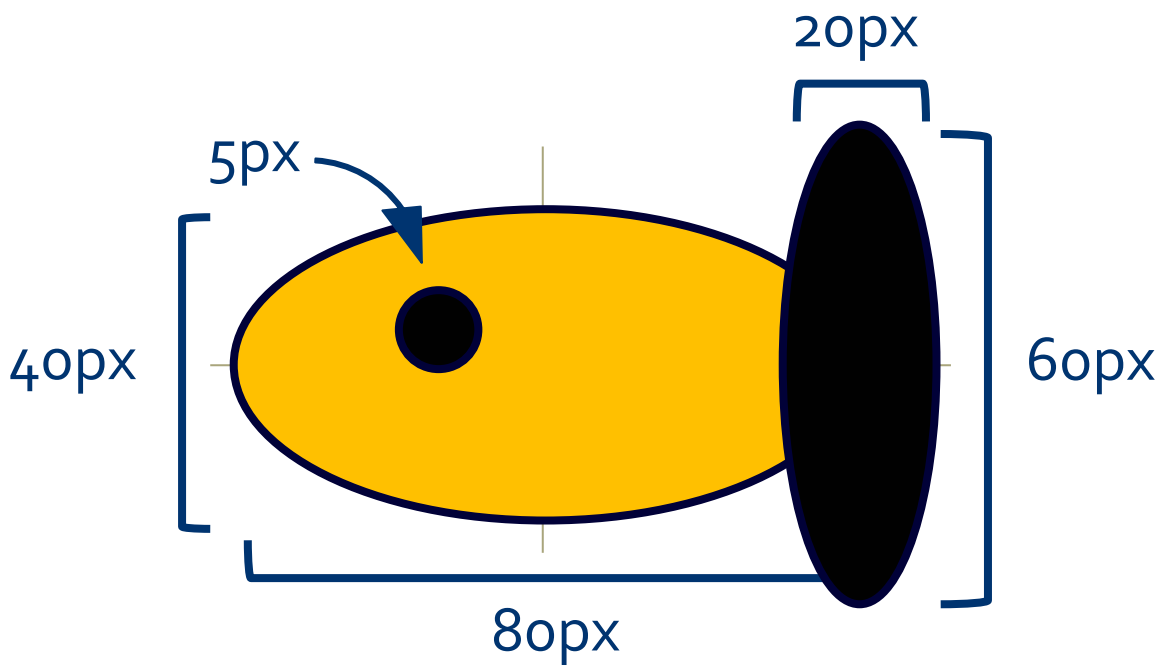
- Project proposal check-in
- Interaction basics
  - mouse
  - keyboard

# User Centered-Design and Prototyping

- Lingering Questions?

# Proposal Check-in

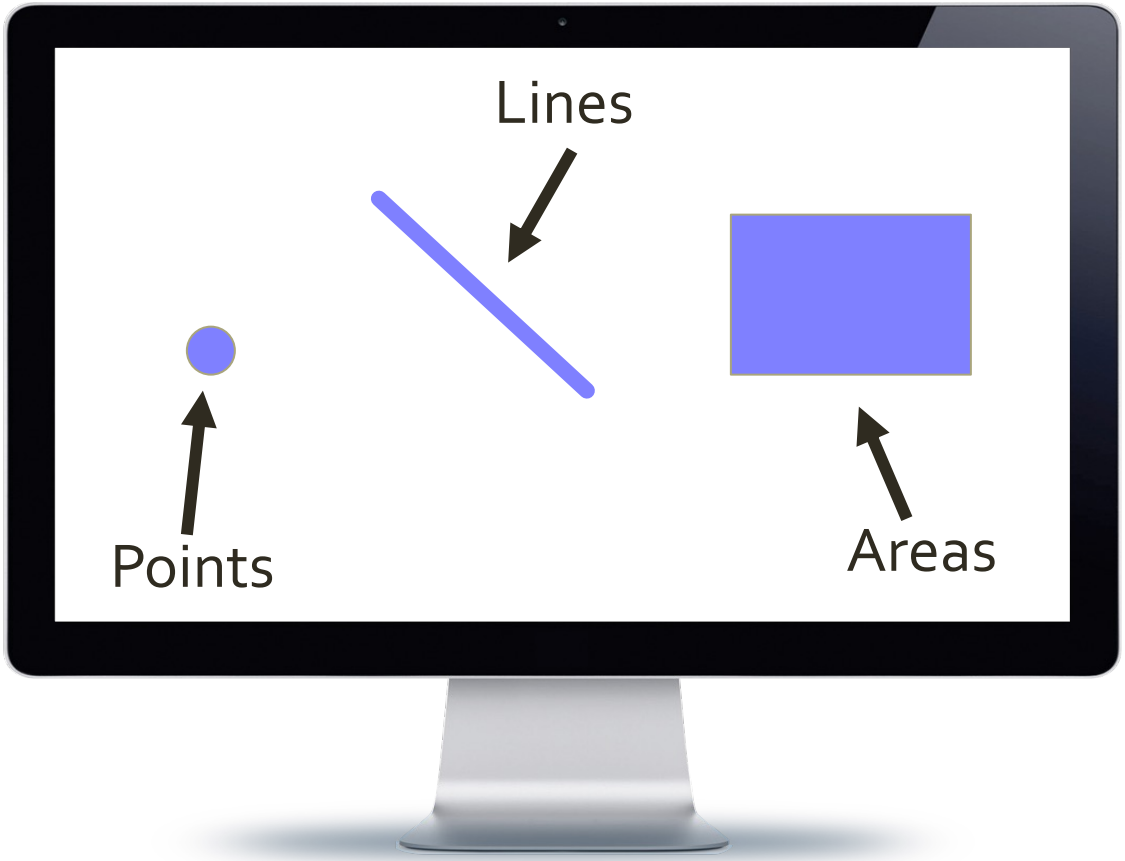
- Work on the animated fish from last class while I come around to check-in on project proposals




- Fish should swim across the screen and flip directions when they hit the edge

- ✓ Draw stuff

"graphical primitives"



- ✓ Draw stuff

using the **`graphics`** module

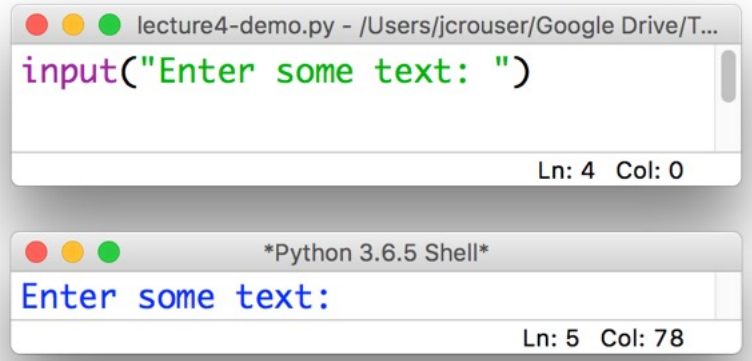
- ✓ Make it move

# 3. Get input from the user and react

# Lecture 4: first experience with user input

## The `.input()` function

- Python has a built-in `.input()` function that allows us to ask the user to type in information
- The `.input()` function takes in a value, which will be printed to the console as a prompt:



# Interaction (def.)

- Ways for the user to **affect change** in what's happening in the program

# Interaction (def.)

- Ways for the user to **affect change** in what's happening in the program

- Low level: **between human and interface**
  - the set of operations available
  - happens between the human and the physical computer

# Interaction (def.)

- Ways for the user to **affect change** in what's happening in the program

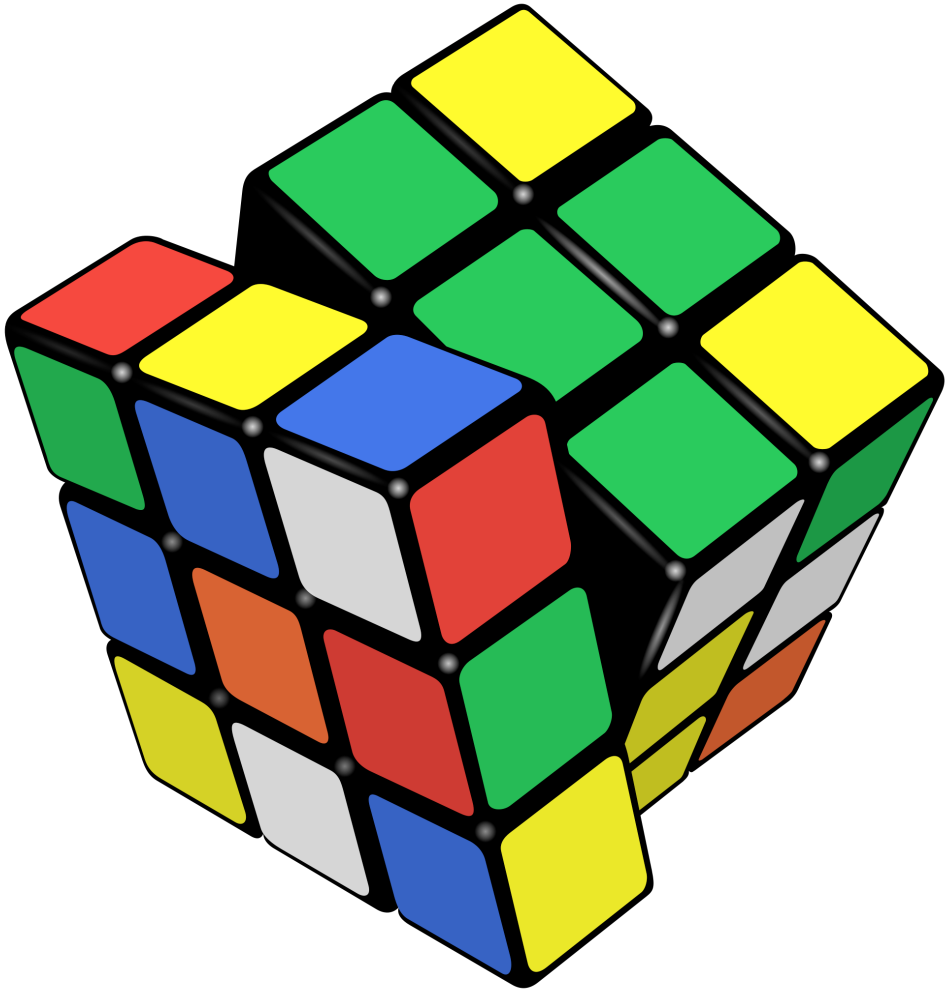
- Low level: **between human and interface**
  - the set of operations available
  - happens between the human and the physical computer

- High level: between **human and problem space**
  - a cognitive act *enabled* by the interface
  - happens between the human and the digital objects

# Example: Rubik's Cube

What **low-level** interactions can you have?

What **high-level** interactions can you have?



# Low-level vs. high-level interactions

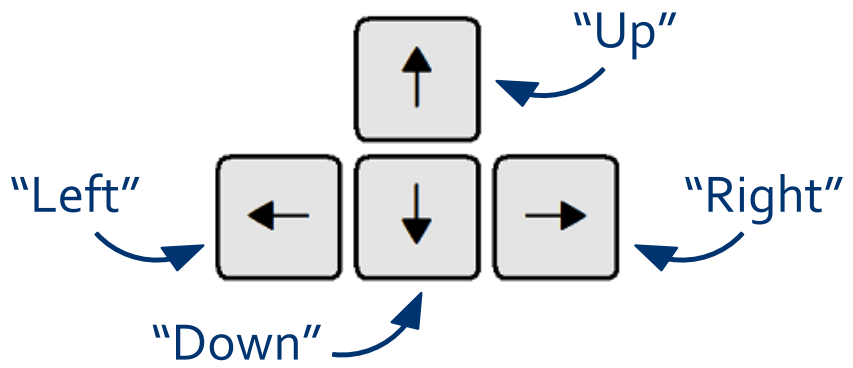
# Interaction with `graphics` objects

- The **`GraphWin`** object has methods to detect interactions
- Mouse:
  - **`.getMouse()`** : stop the program and wait for user to **click**
  - **`.checkMouse()`** : continuously check if the user has **clicked**
  - both return a **`Point`** object
- Keyboard:
  - **`.getKey()`** : stop the program and wait for user to **type**
  - **`.checkKey()`** : continuously check if the user has **typed**
  - both return a **`string`**

Our first interactive **`graphics`** program

# Notes about keyboard interaction

- The strings returned by the **`.getKey()`** / **`.checkKey()`** methods ae called **keycodes**
- Some keys don't have an obvious letter attached to them, but their keycodes are still pretty intuitive, e.g.




- See also: **"space", "Escape", "minus", "underscore", "equal", "plus", "BackSpace", "Return", etc.**

# Back to the Fish Tank

- Start with your fish from the last two classes
- Do the following…

# Challenge 1: press ‘q’ to quit

# Challenge 2: fish position

# Challenge 3: fish frenzy

# Activity: Fish Tank

- Challenge 1: Quit when the user presses "q"
- Challenge 2: Add a fish wherever the user clicks
- Challenge 3: If the user presses the space bar, have all the fish swim to the nearest edge of the screen