Classes & Objects

A class combines (and abstracts) data and functions

An object is an instantiation of a class
class Ball:
    def __init__(self, start_x, start_y, start_v_x, start_v_y, color='blue'):
        # Ball location and velocity
        self.x = start_x
        self.y = start_y
        self.v_x = start_v_x
        self.v_y = start_v_y

        # Ball color, for drawing purposes
        self.color = color

    def update_position(self, timestep):
        self.x = self.x + timestep * self.v_x
        self.y = self.y + timestep * self.v_y

    def update_velocity(self, timestep):
        self.v_y = self.v_y + timestep * EARTH_GRAVITY_ACCELERATION

    def animate_step(self, timestep):
        self.update_position(timestep)
        self.update_velocity(timestep)

    def draw(self):
        D.append(draw.Circle(self.x, self.y, BALL_RADIUS, fill=self.color))
```python
# Drill: BankAccount

class BankAccount:
    def __init__(self, initial):
        self.balance = initial

    def deposit(self, amount):
        self.balance = self.balance + amount

    def withdraw(self, amount):
        self.balance = self.balance - amount

    def overdrawn(self):
        return self.balance < 0

    def __str__(self):
        return "balance: " + str(self.balance)

# Driver code to test BankAccount
my_account = BankAccount(150)
my_account.deposit(200)
p = print( my_account )
```
# Drill: Kangaroo

class Kangaroo:
    def __init__(self):
        self.pouch_contents = []

    def put_in_pouch(self, x):
        for i in range(len(self.pouch_contents)):
            if self.pouch_contents[i] == x:
                print("already in pouch")
                return
        self.pouch_contents.append(x)

    def __str__(self):
        if len(self.pouch_contents) == 0:
            return "The kangaroo's pouch is empty."
        else:
            return "The kangaroo's pouch contains: " + str(self.pouch_contents)

K = Kangaroo()
print(K)
K.put_in_pouch( "ball" )
print(K)
K.put_in_pouch( "hammer" )
print(K)
K.put_in_pouch( "ball" )
print(K)
angle = \tan^{-1}\left( \frac{ (l_{y}-y) }{ (l_{x}-x) } \right)
\[ \cos(\text{angle}) = \frac{x}{r} \]
\[ x = r \times \cos(\text{angle}) \]
Eyes

\[
\sin(\text{angle}) = \frac{y}{r}
\]

\[
y = r \times \sin(\text{angle})
\]
Write a “Crowd” class:

The constructor should take as input a number N and create a list of N “Faces”

The member function lookat should have each face look at a random location

The member function draw should draw all of the faces

In my solution, I made my canvas 800x800, N=25, with a face size of 100 (as before).
# --- DRILL ----
# Write a “Crowd” class:
#
# The constructor should take as input a number N
# and create a list of N “Faces”
#
# The member function lookat should have each face
# look at a random location
#
# The member function draw should draw all of the
# faces
#
# In my solution, I made my canvas 800x800, N=25,
# with a face size of 100 (as before).

[ lecture08-2.ipynb ]