Python OOP Intro

A **class** is a collection of related **objects**, including **attributes** (like characteristics) and **methods** (like actions).

Naban, Dean of Iteration
1
Legendary Creature — Human Wizard
If a Wizard entering the battlefield under your control causes a triggered ability of a permanent you control to trigger, that ability triggers an additional time.
"Perfect. Now do it again."
2/1
058/269 R
DOM • EN Ryan Alexander Lee
™ & © 2018 Wizards of the Coast

Jodah, Archmage Eternal
1
Legendary Creature — Human Wizard
Flying
You may pay rather than pay the mana cost for spells that you cast.
"Chronicles across the ages describe Jodah. They likely refer not to one mage, but to a family or an arcane title."
—Arkol, Argivian scholar
4/3
198/269 R
DOM • EN Yongjae Choi
™ & © 2018 Wizards of the Coast

```
class Wizard:
    name = "Defaultanor" # set attribute, ‘name’ for the class ‘Wizard’
    creature = “Human Wizard”
    attire = [“Cape”, “Robes”, “Staff”]

    def change_name(self, new_name): # note that the first argument is self
        self.name = new_name # access the class attribute with the self keyword
```

You can instantiate the class to assign its attributes and methods to a variable (player1):

```
player1 = Wizard() # instantiate the class
print(player1.name) # print the current object name

player1.change_name("Novador") # change the name using the change_name method
print(player1.name)
```

You can also define attributes at runtime using the init method

```
class Wizard:

    def __init__(self, name):
        self.name = name

    def change_name(self, new_name):
        self.name = new_name

# two variables are instantiated
naban = Wizard("Naban")
jodah = Wizard("Jodah")

# print the names of the two variables
print(naban.name)
print(jodah.name)
```

Maybe you want to define a number of attributes at once:

```
class Wizard:

    def __init__(self, name, talent, cost_white, cost_red, cost_black, cost_blue,
cost_green, cost_gen, power, toughness):
        self.name = name
        self.cost = [cost_white, cost_red, cost_black, cost_blue, cost_green,
cost_gen]
        self.talent = talent
        self.power = power
        self.toughness = toughness

# two variables are instantiated
naban = Wizard("Naban", "Looping", 0, 0, 0, 1, 0, 1, 1, 2)
jodah = Wizard("Jodah", "Flying", 1, 1, 0, 1, 0, 1, 4, 3)

print(naban.name, naban.talent, naban.cost)
print(jodah.name, jodah.talent, jodah.cost)
```

```
Practice:

Add a new wizard.

Stage a wizard fight.
```