

Python OOP Intro

A **module**, also known as a **namespace**, is a Python file that contains a collection of variable names, known as attributes. It's like a library of tools you can access by importing the .py file.

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



```
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

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)

time.sleep(5) #going to need to import the time module to use this

print("FIGHT!")

time.sleep(2)
print("BIFF!      ", "\U0001F4A5")

time.sleep(2)
print("BAM!      ", "\U0001F4A5")

time.sleep(2)
print("\U0001F4A5", "      POW!")

time.sleep(5)

if (naban.toughness <= jodah.power):
    print("Jodah Wins")
else:
    print("Naban Wins")

```

Now add one of these wizards to the battle. How would you stage a wizard fight?



Your turn. Make another class for these characters:

Dragonlord Dromoka 4



Legendary Creature — Elder Dragon

Dragonlord Dromoka can't be countered.
Flying, lifelink
Your opponents can't cast spells during your turn.

Dromoka's followers forsake blood ties so that they may join a greater family.

5/7

217/264 M
DTK • EN • ERIC DESCHAMPS

Dragonlord Silumgar 4



Legendary Creature — Elder Dragon

Flying, deathtouch
When Dragonlord Silumgar enters the battlefield, gain control of target creature or planeswalker for as long as you control Dragonlord Silumgar.

Silumgar never passes up an opportunity to add to his opulence.

3/5

220/264 M
DTK • EN • STEVEN BELLEDIN

Acid-Spewer Dragon 5



Creature — Dragon

Flying, deathtouch
Megamorph 5 (You may cast this card face down as a 2/2 creature for 3. Turn it face up any time for its megamorph cost and put a +1/+1 counter on it.)
When Acid-Spewer Dragon is turned face up, put a +1/+1 counter on each other Dragon creature you control.

3/3

086/264 U
DTK • EN • JAMES ZAPATA

Balefire Dragon 5



Creature — Dragon

Flying
Whenever Balefire Dragon deals combat damage to a player, it deals that much damage to each creature that player controls.

*If it comes for you, die boldly or die swiftly—
for die you will.*

6/6

Eric Deschamps

Wizard Replica

3



Artifact Creature — Wizard



Flying

♦, Sacrifice Wizard Replica: Counter target spell unless its controller pays 2.

It responds with unnatural precision.

Carl Critchlow

© 1993-2003 Wizards of the Coast, Inc. 275/306

1/3