**How to create your own class in Python?**

Empty Class:

class User:

user1 = User()

Will give error. As Python expects indentation after a Class/Function.

To leave a class/ function empty, we use a keyword **pass**

class User:

    pass

user1 = User()

Naming a class: Pascal Casing.

Note:

PascalCase, camelCase, snake\_case

**Working with Attributes, Class Constructors and the \_\_init\_\_() Function:**

Adding attributes:

An attribute is a variable that is associated with an object.

user1 = User()

user1.id = "001"

user1.username = "firdous"

print(user1.username)

However, creating attributes this way is a very long and complex process and we will be creating attributes in another way:

CONSTRUCTOR: A blueprint that allows us to specify what should happen when an object is created.

This is also known as initializing an object.

In Python, we create a structure by using a special FUNCTION called the \_\_init\_\_() function.

Init function will be called every time we create an Object for our class.

class User:

    def \_\_init\_\_(self):

        print("New user being created...")

user1 = User()

user1.id = "001"

user1.username = "firdous"

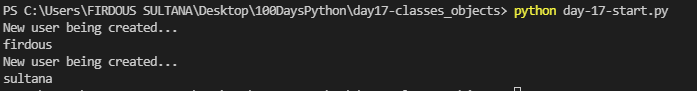
print(user1.username)

user2 = User()

user2.id = "002"

user2.username = "sultana"

print(user2.username)



Parameters passed inside an init function:

self – for actual object

other parameters…

class Car:

    def \_\_init\_\_(self, seats):

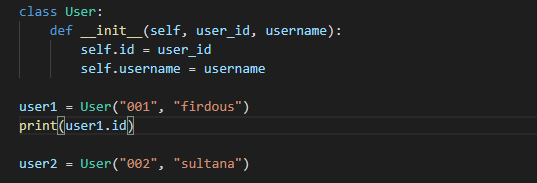
        self.seats = seats

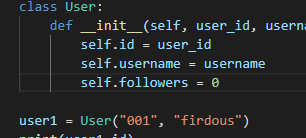
my\_car = Car(5)

This is a lot quicker way than doing my\_car.seats = 5

(When we need to create a lot of objects that has same Attributes)

So, we have:



We can also have default attributes for our class. E.g.:  self.followers = 0

**Adding methods to a Class**:

Attributes – Things that an object HAS

Methods – Things that and Object DOES

Say, in Car class; when mode has to change to the raceing car mode, the seats of the car/ truck will be reduced( to make the vehicled light to run faster)

class Car:

def enter\_race\_mode():

self.seats = 2

class User:

    def \_\_init\_\_(self, user\_id, username):

        self.id = user\_id

        self.username = username

        self.followers = 0

        self.following = 0

    def follow(self, user):

        user.followers += 1

        self.following +=1

user1 = User("001", "firdous")

user2 = User("002", "sultana")

user1.follow(user2)

print(user1.followers)

print(user1.following)

print(user2.followers)

print(user2.following)

Simple true false Quiz Game…