Murdeus (aka.Murat Can Demiral) Follow

Dec 15, 2022 · 4 min read · Listen

Save

How To Learn Classes in Python About 4 Minutes

Almost everything in Python is a class. A class is an object (object) that can instantiate itself.Classes can be assigned for properties and purposes.A class is a comprehensive code combination of objects. In OOP (Object Oriented Programming), the object is the small building block that contains the properties of the class. So when we write a class, we gather the properties and methods of various objects together.

Syntax: Class Definition

```
class ClassName:
    # Statement
```

Syntax: Object Definition

```
obj = ClassName()
print(obj.atrr)
```

Some points on Python class:

- Classes are created by keyword class.
- Attributes are the variables that belong to a class.

- Attributes are always public and can be accessed using the dot (.) operator. Eg.: Myclass.Myattribute

Defining a class

```
# Python3 program to
# demonstrate defining
# a class


class Dog:
    passppp
In the above example, the class keyword indicates that you are
creating a class followed by the name of the class (Dog in this case).
```

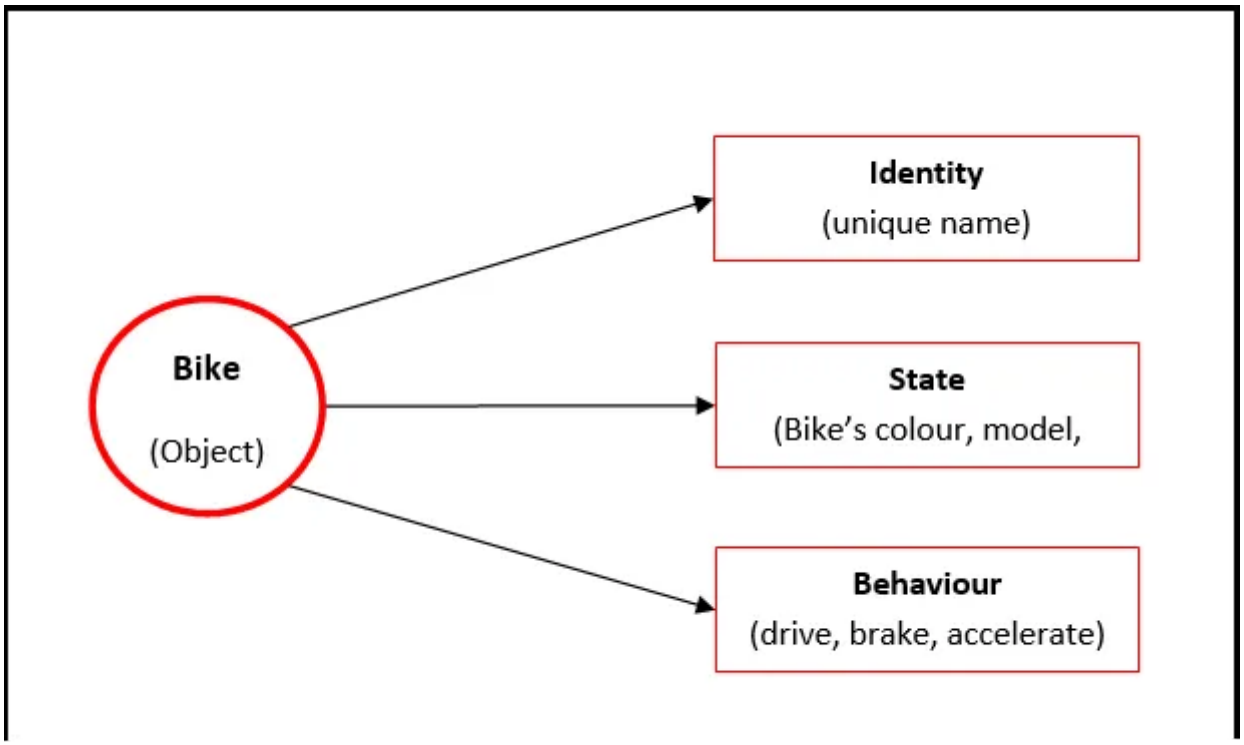
Class Objects

(Object): A unique instance of a data structure defined by its class. An object contains both data members (class variables and instance variables) and methods.

Operator overloading: Assigning more than one function to a specific operator.

An object consists of :

- **State:** It is represented by the attributes of an object. It also reflects the properties of an object.
- **Behaviour:** It is represented by the methods of an object. It also reflects the response of an object to other objects.
- **Identity:** It gives a unique name to an object and enables one object to interact with other objects.



-

Declaring an object

```
# demonstrate instantiating
# a class

class Cat:
    # A simple class
    # attribute
    attr1 = "mammal"
    attr2 = "Cat"

    # A sample method
    def fun(self):
        print("I'm a", self.attr1)
        print("I'm a", self.attr2)

# Driver code
# Object instantiation
Blacknight = Cat()

# Accessing class attributes
# and method through objects
print(Blacknight.attr1)
Blacknight.fun()
```

Output

```
mammal
I'm a mammal
I'm a Cat
```

The self

The self is a concept that comes with the __init__ method and enables us to achieve the goal we have derived from the class.

- Class methods must have an extra first parameter in the method definition. We do not give a value for this parameter when we call the method, Python provides it.
- If we have a method that takes no arguments, we still have one argument.
- This is similar to this pointer in C++ and this reference in Java.

__init__ method

_init__ is a class's constructor method in OOP programming. If we are going to derive an object from a class, __init__ has to be the first method of the class. Properties of objects derived from Class are assigned to objects with this method.

```
class Cat:
   # init method or constructor
    def __init__(self, p_name, ):
        self.name = p_name
    # Sample Method
    def mycat(self):
        print('My cats name', self.name)

p = Cat("Blacknight")
p.say_mycat()
```

Output:

```
My cats name Blacknight
```

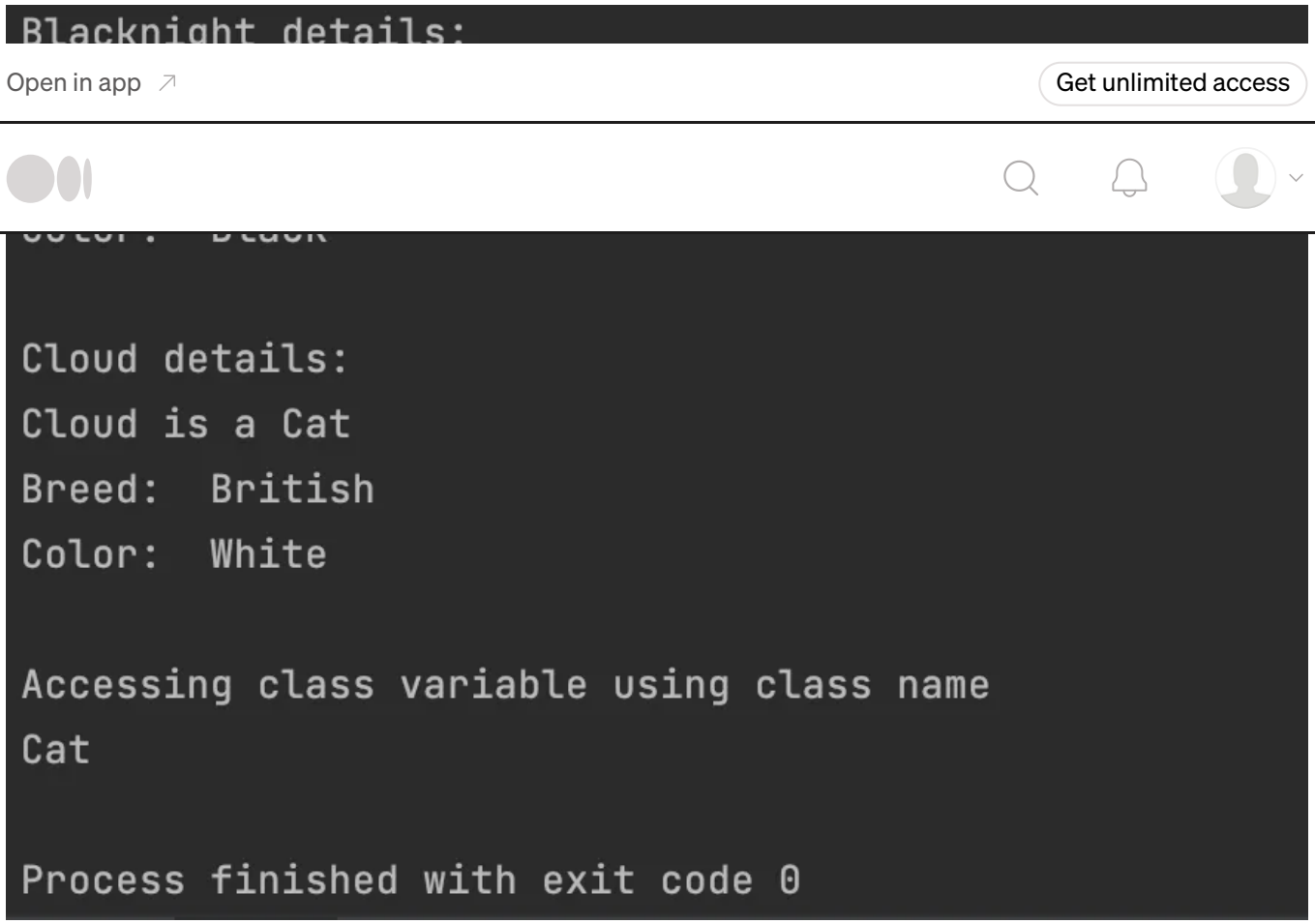
Class and Instance Variables

Instance variables are for data, unique to each instance and class variables are for attributes and methods shared by all instances of the class. Instance variables are variables whose value is assigned inside a constructor or method with self whereas class variables are variables whose value is assigned in the class.

Defining instance variables using a constructor.

```
class Cat:
    # Class Variable
    animal = 'Cat'

    # The init method or constructor
    def __init__(self, breed, color):
        # Instance Variable
        self.breed = breed
        self.color = color


# Objects of Dog class
Blacknight = Cat("Scottish", "Black")
Cloud = Cat("British", "White")

print('Blacknight details:')
print('Blacknight is a', Blacknight.animal)
print('Breed: ', Blacknight.breed)
print('Color: ',Blacknight.color)

print('\nCloud details:')
print('Cloud is a', Cloud.animal)
print('Breed: ', Cloud.breed)
print('Color: ', Cloud.color)

# Class variables can be accessed using class
# name also
print("\nAccessing class variable using class name")
print(Cat.animal)
```

Output:



And dont forget"You failed at a thing cause you need to succeed at it. Failure gives meaning to success. Don't dare give up!"

Just someting i read i hope it makes ur day good.

See Ya.

Resource:

https://www.geeksforgeeks.org/python-classes-and-objects/amp/

Python Software Developer