

## Instance variable

Any variables which we make inside Constructor are known as instance variable.

Why → Because value of these variables for each objects are different.

## Encapsulation

Whenever we create any class, it is not good practice to keep open all data to get accessed by anyone or everyone.

Solution is to hide the data

To make private any variable we use double underscore in front of that variable :   
Self.\_pin = ""  
Self.\_balance = 0

Concept 1 : If you are creating a class then it is a good practice to hide your class data members. If you want, you can hide any method also if you don't want to show that to others.

## Behind the Scene what happens (How Python able to hide)

When we use double underscore in front of any variable, Python interpreter internally while execution replaces that variable with \_class\_variable . For example :

\_pin → \_Aton\_pin

If we do object.\_variable it will create new variable, and don't disturb the existing logic.

We can access these private or protected variables by

object.\_class.\_variable

So we can say that nothing in python is truly private. because we can access those private variables by this

## Syntax.

Why in Python we don't able to keep anything truly private?

→ In python, you can not hide anything completely because Python is a language made for adults, it means you should have enough maturity to know that if any variable is being protected by double underscore (--) then you should not touch that. It is like a gentle man agreement that if I have kept something private by double underscore then any programmer should not touch that.

And in future if you need you also able to access that private variable by above Syntax. (Python said)

Python does not support completely hiding technique like JAVA. In JAVA if you write Private in any variable then that variable will become private Completely.

Way to give access to private variable :

Getter Method and Setter Method

The whole idea is, first hiding your data and then creating these two methods and allowing access and change in private variable by the help of these two methods, so that whatever change will happen that should satisfy our logic written on those two methods.

So it means whatever will pass through those method, should first go through our logic and if it is true then only change will happen.

So in this way we are not giving direct access to private variables and only allowing those changes we decide in our logic by the help of those getter and setter method.

### Two concepts

(i) first we hide the data

(ii) If user want we can give access but using getter and setter method.

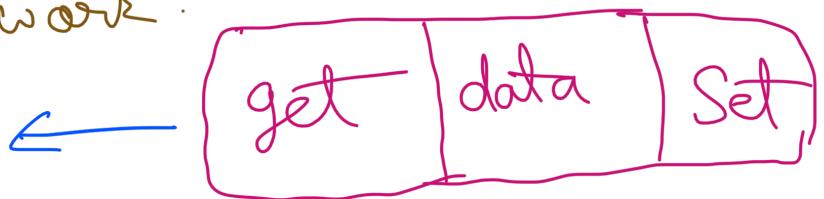
getter method → for fetching the value

setter method → for setting value but according our rule.

So in a way, we are protecting our data if you want to see data, See with the help of function and if you want to set something, Set by going through my logic

And that is the whole idea behind Encapsulation, for each data member, we are making two method getter and setter. And these 3 together (data, getter, setter) has to make data work.

Encapsulation



Class diagram

