Object oriented programming:

In traditional method of programming (procedure oriented) programming, data and functions are separate or we can say different enity. Will send some data to a function and it will return some value.

But function(procedure) cann’t hold any data, it’s a drawback.

Where as in object oriented programming data & method/function are part of a single entity called class or we can say it is bluprint. So we can create a object/instance of a class and use its members

|  |  |
| --- | --- |
|  |  |

Note:

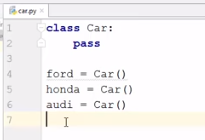
Data in class called attribute or member variables.

Function in class called method.

Classes and objects in python:

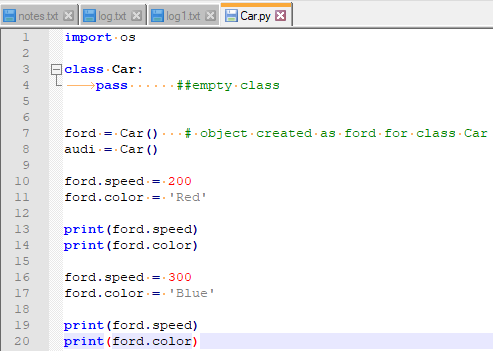
We can Create class using class keyword

Example car.py:

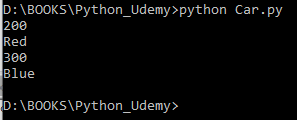


**pass** means empty class, we can create empty function/method as well.

Ford,Honda & audi are object of class Car.



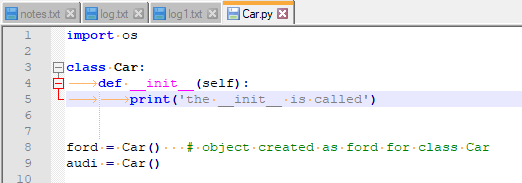
O/p-



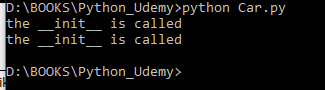
Python \_\_init\_\_ and self in class:

\_\_init\_\_ method is like a constructor and first method to called when any object will be created for the class , where \_\_init\_\_ method present. And self will the first argument for the \_\_init\_\_ method.

Please check the below example:



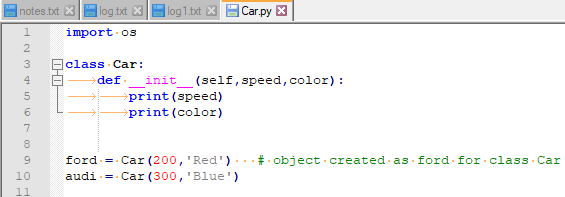
O/p-

 twice called .. when two times object created

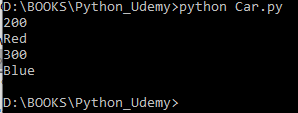
Note: We called \_\_init\_\_ as a constructor but it is not a actual constructor but behave like a constructor.

And there is no destructor in python as it has facility of automatic garbage collection.

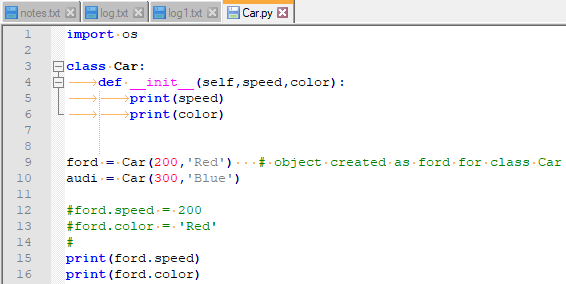
As \_\_init\_\_ method we can use for initialization, we can use the above programme like below as well:



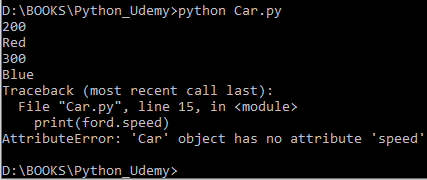
O/P-



Example:



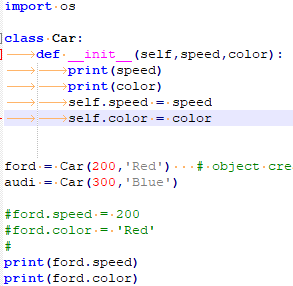
O/P-



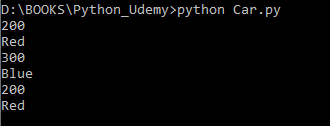
As there is no value provide to speed and color attribute. So we are seeing this error..

Follow below procedure to avoid the error:

Self.speed and self.color



O/P-

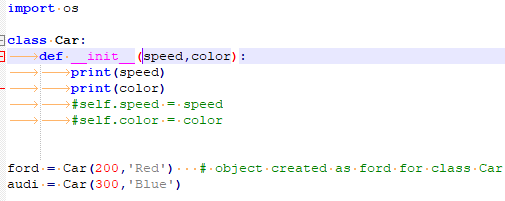


Note:

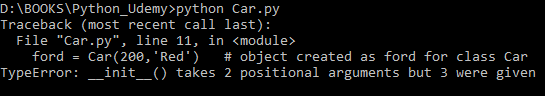
We use ‘self’ as first argument for every method , but it is not necessary that it will ‘self’ only.

We can give some other name as well.

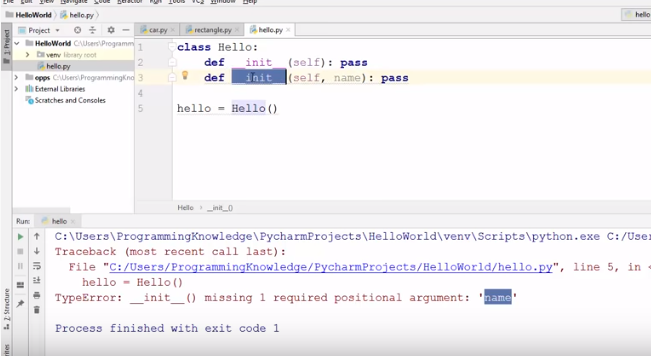
With out self argument:



o/p-



Multiple \_\_init\_\_ method:

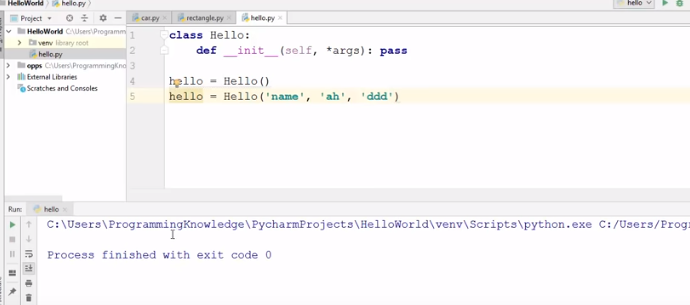


Note:

Multiple \_\_init\_\_ method is not possible in a class , but if you give then last \_\_init\_\_ method will be the final one.

How to provide multiple arguments?

Ans> by using \*args



Note:

For dictionary we can provide \*\*kwargs in \_\_init\_\_ method and in object creation we can pass key,value pair like below:  
hello = Hello(name=’bibhu’)

Python Encapsulation:

If for a attribute which takes numeric value .., but if I will give string for that .. then it will work but not a valid one.

So we need to provide security to our data to be modified so easily.

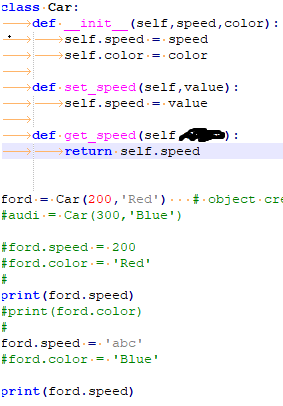
This is called encapsulation , which provides security to our data to be modified.

We can use get\_ & set\_ methods to secure data from direct modification.

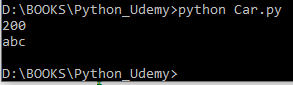
As we know in c++ there is a concept of private,public & protected keywords to protect our data.

But in python there is now such keyword .

In python for private we use double underscore(\_\_) before any attribute.

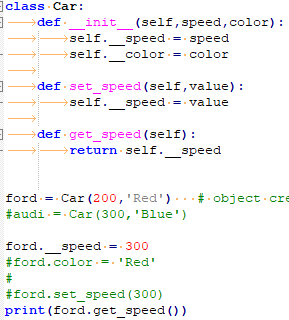
 Her also we can use ford.set\_speed(400) to set the value and ford.get\_speed() to get the speed value.

O/P-

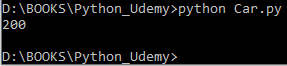


We can see in the above example after using set\_ and get\_ also our data can be modified.

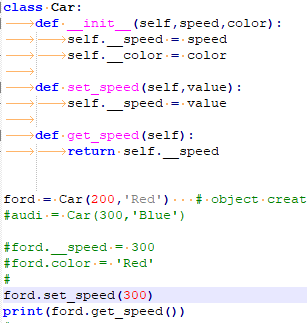
So now will use underscore(\_\_) before any attribute to make our data private.

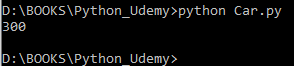


O/P-

ford.\_\_speed() is not working as it is private.

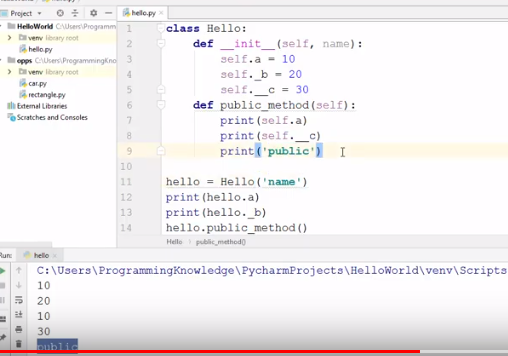
But we can change private value using set\_speed() method.



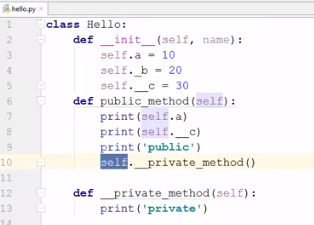


Note:

Private member variables are not accessible outside of class, but accessible in the same class with different member function.



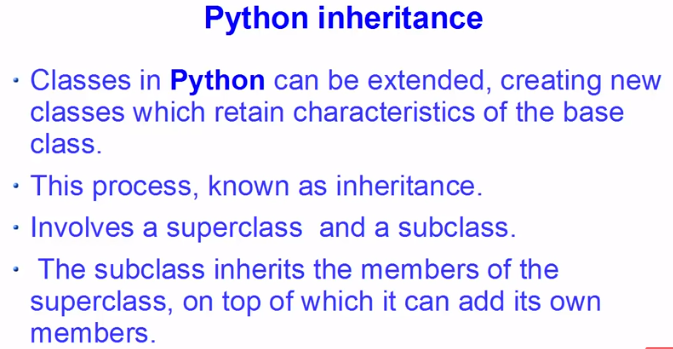
Private Method:



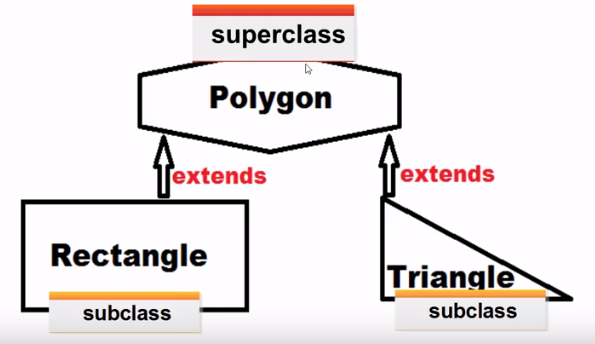
Note:

Private method cann’t called outside class , but can be called outside class.

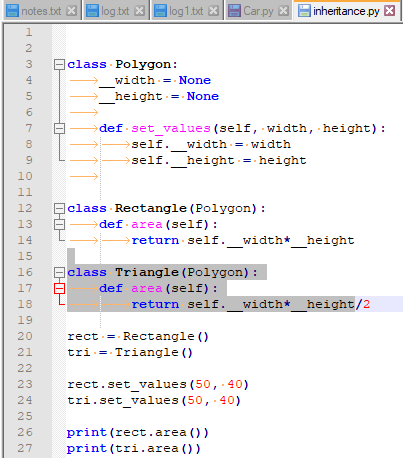
Inheritance in python:



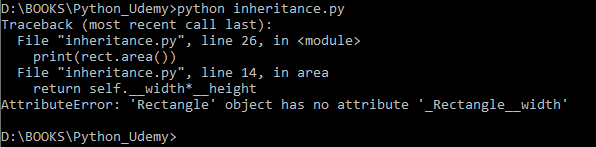
Example:



Example:



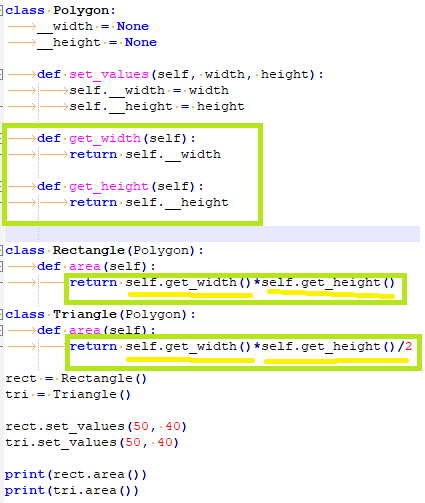
O/P-



Note:

\_\_width & \_\_height are private members of polygon superclass so it cannot be accessible to subclass.

Solution:



O/P-

