Q1. What is the difference between __getattr__ and __getattribute__?

Answer : __getattribute__ is used to find an attribute of a class. It raises an AttributeError of it fails to find an attribute of a class. __getattr__ is implemented latter if AttributeError is generated by __getattribute__ , but for this __getattribute__ and __getattr__ both has to be defined in same class. If no attribute is found, __getattr__ returns a default value. So key difference is that __getattr__ is called for attributes that don't actually exist on a class.

Q2. What is the difference between properties and descriptors?

Answer :

Properties: We can bind getter, setter and delete functions with an attribute name, using the built-in property function. When we do this, each reference to an attribute looks like simple, direct access, but invokes the appropriate function of the object.

Descriptors: We can bind getter, setter (and deleter) functions into a separate class. We then assign an object of this class to the attribute name. When we do this, each reference to an attribute looks like simple, direct access, but invokes an appropriate function of the Descriptor object.

Q3. What are the key differences in functionality between __getattr__ and __getattribute__, as well as properties and descriptors?

Answer :

getattr : It return the value of attributes which is constructed inside the class and having an class object which store its value.

If the attribute is not present inside the class and if default value is provided in the syntax only then it will return that value else it will show attribute error

getattribute : This method will invoked before looking at the actual attributes on the object. Means, if ywe have getattribute method in our class, python invokes this method for every attribute regardless whether it exists or not.

Properties: We can bind getter, setter and delete functions **with an attribute name, using the built-in property function.** When we do this, each reference to an attribute looks like simple, direct access, but invokes the appropriate function of the object.

Descriptors: We can bind getter, setter (and deleter) functions **into a separate class. We then assign an object of this class to the attribute name.** When we do this, each reference to an attribute looks like simple, direct access, but invokes an appropriate function of the Descriptor object.