Q1. What is the difference between \_\_getattr\_\_ and \_\_getattribute\_\_?

Ans : getattribute: Is used to retrieve an attribute from an instance.

getattr: Is executed as the last resource when attribute is not found in an object.

\_\_getattr\_\_ vs \_\_getattribute\_\_

These two methods are implemented as methods of a class. \_\_getattribute\_\_ is always called. Essentially this method is used to find an attribute of a class. If it fails, it will raise an AttributeError. In case it fails, and class implements \_\_getattr\_\_, then \_\_getattr\_\_ is called right after. Therefore, the biggest 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?

Ans :

Descriptors

A Descriptor is a class which provides detailed get, set and delete control over an attribute of another object. This allows you to define attributes which are fairly complex objects in their own right. The idea is that we can use simple attribute references in a program, but those simple references are actually method functions of a descriptor object.

Properties

The property function gives us a handy way to implement a simple descriptor without defining a separate class. Rather than create a complete class definition, we can write getter and setter method functions, and then bind these functions to an attribute name.

Q3. What are the key differences in functionality between \_\_getattr\_\_ and \_\_getattribute\_\_, as well as properties and descriptors?

Ans : These two methods are implemented as methods of a class. \_\_getattribute\_\_ is always called. Essentially this method is used to find an attribute of a class. If it fails, it will raise an AttributeError. In case it fails, and class implements \_\_getattr\_\_, then \_\_getattr\_\_ is called right after. Therefore, the biggest difference is that \_\_getattr\_\_ is called for attributes that don’t actually exist on a class.

descriptors are a low-level mechanism that lets you hook into an object's attributes being accessed. Properties are **a high-level application** of this; that is, properties are implemented using descriptors.