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

\_\_getattr\_\_ is only invoked if the attribute wasn't found the usual ways. It's good for implementing a fallback for missing attributes, and is probably the one of two you want. \_\_getattribute\_\_ is invoked before looking at the actual attributes on the object, and so can be tricky to implement correctly. You can end up in infinite recursions very easily.

Q2. What is the difference between properties and descriptors?

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

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

difference between \_\_getattribute\_\_() and \_\_getattr\_\_() is that the first one is called unconditionally when an attribute is being retrieved from an instance while the second is called only when the attribute was not found

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.