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

**\_getattribute\_\_ has a default implementation, but \_\_getattr\_\_ does not**. This has a clear meaning: since \_\_getattribute\_\_ has a default implementation, while \_\_getattr\_\_ not, clearly python encourages users to implement \_\_getattr\_\_

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?

 It 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**.

 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.