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

Ans1

Difference between the two is that \_\_getattr\_\_ is only called when an attribute lookup fails, while \_\_getattribute\_\_ is always called. This means that \_\_getattribute\_\_ has more control over attribute access, but it can also be more dangerous, as it can potentially create an infinite loop if try to access an attribute within the \_\_getattribute\_\_ method. Therefore, it is important to be cautious when using \_\_getattribute\_\_.

Q2. What is the difference between properties and descriptors?

Ans2

Difference between properties and descriptors is the level of abstraction they provide. Properties are a high-level way to customize attribute access, while descriptors are a more low-level way to customize attribute access. Properties are easier to use and understand, but they have some limitations. Descriptors are more flexible and powerful, but they require more understanding and are harder to use correctly.

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

Ans3

|  |  |
| --- | --- |
| \_\_getattr\_\_ | \_\_getattribute\_\_ |
| \_\_getattr\_\_ is called when an attribute is not found in the usual way, meaning when you try to access an attribute that doesn't exist in the object. It is typically used to handle missing attributes and return a value or raise an AttributeError. | \_\_getattribute\_\_ is called every time an attribute is accessed, whether it exists or not. It is typically used to customize the behavior of attribute access, such as logging attribute accesses or dynamically computing attribute values. |
| Properties | Descriptors |
| Properties provide a way to customize attribute access by defining methods that can be used to get, set, and delete the value of an attribute, but from the outside, they look like regular attribute access. Properties are often used to provide read-only or write-only access to an attribute, or to perform additional calculations when an attribute is accessed. | Descriptors are a lower-level way to customize attribute access. They allow to define methods that get called when an attribute is accessed, set, or deleted. Descriptors are typically used to enforce constraints on attribute values, such as making sure that a value is of a certain type or to implement caching of attribute values. Descriptors are more flexible than properties, as they can be reused with different attributes in different classes. |