

1.

`getattr`: Is used to retrieve an attribute from an instance.

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

2.

Python's `property()` is the Pythonic way to avoid formal getter and setter methods in your code. This function allows you to turn class attributes into properties or managed attributes. Since `property()` is a built-in function, you can use it without importing anything.

Python descriptors are created to manage the attributes of different classes which use the object as reference.

3.

`getattr`: Is used to retrieve an attribute from an instance. `__getattr__`: Is executed as the last resource when attribute is not found in an object.

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.