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

The **\_\_getattr\_\_ and \_\_setattr**\_\_ methods, **for routing undefined attribute fetches**

**and all attribute assignments to generic handler methods**.

• The \_\_getattribute\_\_ method, for **routing all attribute fetches to a generic handler**

**method**

Q2. What is the difference between properties and descriptors?

The **property protocol** **allows us to route a specific attribute’s get, set, and delete operations**

**to functions or methods we provide, enabling us to insert code to be run automatically**

**on attribute access, intercept attribute deletions, and provide documentation**

**for the attributes if desired**

property is created by assigning the result of a built-in function to a class attribute:

attribute = property(fget, fset, fdel, doc)

class Person: *# Add (object) in 2.X*

def \_\_init\_\_(self, name):

self.\_name = name

def getName(self):

print('fetch...')

return self.\_name

def setName(self, value):

print('change...')

self.\_name = value

def delName(self):

print('remove...')

del self.\_name

name = property(getName, setName, delName, "name property docs")

bob = Person('Bob Smith') *# bob has a managed attribute*

print(bob.name) *# Runs getName*

bob.name = 'Robert Smith' *# Runs setName*

print(bob.name)

del bob.name *# Runs delName*

print('-'\*20)

sue = Person('Sue Jones') *# sue inherits property too*

print(sue.name)

print(Person.name.\_\_doc\_\_)

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