**Write about the following built-in attributes in OOP**

**#\_\_dict\_\_**

* It is as special attribute of every module.
* This is the dictionary containing module’s symbol table.
* It will return a dictionary object of module attribute, functions and other definitions and their respective values.
* E. g:

import math

print(math.\_\_dict\_\_)

**#\_\_doc\_\_**

* It denotes the document string line written in module code.
* It returns the doc string defined in the function.
* E. g:

def demo():

'''

This is an example of how a doc\_string looks like.

This string give useful information about the function being defined.

'''

print(demo.\_\_doc\_\_)

**#\_\_name\_\_**

* It returns the name of the module. by default, the name of the file (excluding the extension.py) is the value \_\_name\_\_ attribute.
* When we run any Python script (i.e. a module), its \_\_name\_\_ attribute is also set to \_\_main\_\_.
* E. g:

print("\_\_name\_\_ = ", \_\_name\_\_)

-It will return \_\_main\_\_.

**#\_\_module\_\_**

* This built-in class attribute when called, print the name of the module the function /object was defined in ,or None if unavailable.
* E.g:

class A(object):

pass

class B(A):

pass

b = B()

print B.\_\_module\_\_

🡪This will give output \_\_main\_\_.

**#\_\_bases\_\_**

* This built in class attribute when called prints the tuple of bae classes of class object.
* The following codes shows how \_\_bases\_\_ works.
* B is a child class of the parent class A.
* E. g:

class A(object):

pass

class B(A):

pass

b = B()

print B.\_\_bases\_\_