Python - Python Class Attributes and Functions

Python में हर class के साथ In-Built Class Attributes होते है |

* **\_\_doc\_\_ :**ये दिए गए class की docstring को return करता है अगर docstring नहीं होती है तो 'None' return होता है |
* **\_\_name\_\_ :**ये दिए गए class का नाम return करता है |
* **\_\_module\_\_ :**जहा पर class होता वो module name return किया जाता है अगर class current program पर होता है तो '\_\_main\_\_' return होता है |
* **\_\_bases\_\_ :**ये दिए गए class का base class(es) tuple में return करता है | अगर कोई base class नहीं होता है तो empty tuple return करता है |
* **\_\_dict\_\_ :**ये दिए गए class के attributes को dictionary में return करता है |

Python में हर class के साथ In-Built Class Attributes होते है |

* **\_\_doc\_\_ :**ये दिए गए class की docstring को return करता है अगर docstring नहीं होती है तो 'None' return होता है |
* **\_\_name\_\_ :**ये दिए गए class का नाम return करता है |
* **\_\_module\_\_ :**जहा पर class होता वो module name return किया जाता है अगर class current program पर होता है तो '\_\_main\_\_' return होता है |
* **\_\_bases\_\_ :**ये दिए गए class का base class(es) tuple में return करता है | अगर कोई base class नहीं होता है तो empty tuple return करता है |
* **\_\_dict\_\_ :**ये दिए गए class के attributes को dictionary में return करता है |

Example for Class Attributes in Python

Source Code :

class Employee:

"I am in class Employee"

pass

class Fitness(Employee):

"I am in class Fitness"

pass

class Company(Fitness):

"I am in class Company"

pass

print("Docstring of class Company :",Company.\_\_doc\_\_)

print("Class Name :",Company.\_\_name\_\_)

print("Module Name :",Company.\_\_module\_\_)

print("Base Class of Company :",Company.\_\_bases\_\_)

print("Company's class attributes :",Company.\_\_dict\_\_)

Output :

Docstring of class Company : I am in class Company

Class Name : Company

Module Name : \_\_main\_\_

Base Class of Company : (<class '\_\_main\_\_.Fitness'>,)

Company's class attributes : {'\_\_module\_\_': '\_\_main\_\_', '\_\_doc\_\_': 'I am in class Company'}

Example for All Class Attributes in Python

Source Code :

class Employee:

"I am in class Employee"

pass

class Fitness(Employee):

"I am in class Fitness"

pass

class Company(Fitness):

"I am in class Company"

pass

obj = Employee()

for attr in dir(Employee):

if attr.startswith('\_\_'):

print(attr)

Output :

\_\_class\_\_

\_\_delattr\_\_

\_\_dict\_\_

\_\_dir\_\_

\_\_doc\_\_

\_\_eq\_\_

\_\_format\_\_

\_\_ge\_\_

\_\_getattribute\_\_

\_\_gt\_\_

\_\_hash\_\_

\_\_init\_\_

\_\_init\_subclass\_\_

\_\_le\_\_

\_\_lt\_\_

\_\_module\_\_

\_\_ne\_\_

\_\_new\_\_

\_\_reduce\_\_

\_\_reduce\_ex\_\_

\_\_repr\_\_

\_\_setattr\_\_

\_\_sizeof\_\_

\_\_str\_\_

\_\_subclasshook\_\_

\_\_weakref\_\_

Functions for Class Attributes

|  |  |
| --- | --- |
|  |  |
| delattr() | object के attribute को delete करने के लिए इस्तेमाल किया जाता है | |
| getattr() | object के attribute की value को return करता है | |
| hasattr() | object का attribute है या नहीं ये check boolean value में return करता है | |
| setattr() | object के attribute के set किया जाता है अगर पहले ही set होता है तो उसे overwrite किया जाता है लेकिन ये अलग-अलग memory share करता है | ये None return करता है | |

delattr() Function in Python

**Syntax :**

delattr(object, attribute)

**Example :**

class MyClass:

a = 5

obj = MyClass()

print(obj.a)

delattr(obj, 'a') #same as del obj.a

#Output :

# delattr(obj, 'a')

#AttributeError: a

getattr() Function in Python

**Syntax :**

getattr(object, attribute)

**Example :**

class MyClass:

a = 5

obj = MyClass()

print(getattr(obj, 'a'))

#same as

print(obj.a)

#Output

#5

#5

hasattr() Function in Python

**Syntax :**

hasattr(object, attribute)

**Example :**

class MyClass:

a = 5

obj = MyClass()

print(hasattr(obj, 'a'))

#same as

print(bool(obj.a))

#Output

#True

#True

print(hasattr(obj, 'b')) #return False if attribute is not set

#not same as

print(bool(obj.b)) #raise AttributeError if attribute is not set

#Output

#False

#AttributeError

setattr() Function in Python

**Syntax :**

setattr(object, attribute, attr\_value)

**Example :**

class MyClass:

a = 5

obj = MyClass()

setattr(obj, 'a', 8)

print(getattr(obj, 'a'))

#Output

#8

setattr(obj, 'b', 10)

print(getattr(obj, 'b'))

#Output

#10