How to Delete object properties and object?

You can delete objects and properties of object by using the del keyword.

Example

```
class Employee:
    def __init__(self, salary, name):
        self.salary = salary
        self.name = name


emp1 = Employee(10000, "John Doe")

del emp1.salary     # Delete object property
del emp1            # Delete object
```

How to check and compare type of an object?

Class, or type, is an object that holds information about how to construct a certain kind of objects and what that kind of objects can do. A type is the class of a class. Like everything else in Python, classes themselves are objects, and you can pass them around, assign them to variables, etc. If you ask a class what its class is, you will get the answer type. If you ask a class instance what its class is, you will of course get the class.

Example

```
class Test(object):
    pass


print(type(Test))

obj1 = Test()
```

```
print(type(obj1))

obj2 = Test()
print(type(obj1) is type(obj2))
```

Output

```
< class 'type' >
< class '__main__.Test' >
True
```

How to copy all properties of an object to another object?

Example

```
class MyClass(object):
    def __init__(self):
        super(MyClass, self).__init__()
        self.foo = 1
        self.bar = 2


obj1 = MyClass()
obj2 = MyClass()

obj1.foo = 25
obj2.__dict__.update(obj1.__dict__)

print(obj1.foo)
print(obj2.foo)
```

Output

```
25
25
```

Print all properties of an Object

Example

```
class Animal(object):
    def __init__(self):
        self.eyes = 2
        self.name = 'Dog'
        self.color= 'Spotted'
        self.legs= 4
        self.age  = 10
        self.kids = 0


animal = Animal()
animal.tail = 1

temp = vars(animal)
for item in temp:
    print(item, ':', temp[item])
```

Output

```
kids : 0
eyes : 2
name : Dog
color : Spotted
tail : 1
legs : 4
age : 10
```

How to create Data Attributes of a class dynamically?

The setattr used to sets the named attribute on the given object with a specified value.

Example

```
class Employee:
    pass


emp1 = Employee()
setattr(emp1, 'Salary', 12000)

emp2 = Employee()
setattr(emp2, 'Age', 25)

print(emp1.Salary)
print(emp2.Age)
```

Output

```
12000
25
```

How to create Private members of class?

If the name of a Python function, class method, or attribute starts with (but doesn't end with) two underscores, it's private; everything else is public.

Example

```
class Test(object):
    __private_var = 100
    public_var = 200

    def __private_func(self):
        print('Private Function')

    def public_func(self):
        print('Public Function')
        print(self.public_var)
```

```
    def call_private(self):
        self.__private_func()
        print(self.__private_var)


t = Test()
print(t.call_private())
print(t.public_func())
```

Output

```
Private Function
100
None
Public Function
200
None
```