1. add() and mul() methods can be used for both integers and string data objects. String is iterable object. When two strings are passed to + operator, it will return concatenated string. When \* (number) is preceeded by string, then that string is repeated those many number of time.

print("abc" + 'def')

print("pqr" \* 5)

2. It depends on the input parameter result is printed. Example, If both inputs are string function will print string output.

3. import profile

import sys

class InterceptedList2(list):

def \_\_setitem\_\_(self, key, value):

print ('saving')

list.\_\_setitem\_\_(self, key, value)

def \_\_delitem\_\_(self, key):

print ('saving')

list.\_\_delitem\_\_(self, key)

l2 = InterceptedList2()

l2.extend([1,2,3,4])

profile.run("l2[3:] = [5]")

profile.run("l2[2:6] = [12,4]")

profile.run("l2[-1:] = [42]")

profile.run("l2[::2] = [6,6]")

4. Python in its definition provides methods to perform inplace operations, i.e doing assignment and computation in a single statement using “operator” module. For example,

x += y is equivalent to x = operator.iadd(x, y)

iadd() function is used to assign and add the current value. This operation does “a+=b” operation. Assigning is not performed in case of immutable containers, such as strings, numbers and tuples.

import operator

x = 2

y = 3

x = operator.iadd(x,y)

print(x)

5. When we have two objects which are a physical representation of a class (user-defined data type) and we have to add two objects with binary ‘+’ operator it throws an error, because compiler don’t know how to add two objects. So we define a method for an operator and that process is called operator overloading.