Day16_list_class

```
lst = [56,98.36,None,6-9j,True,56,"Hello"]
type(lst)
#list

print(type(lst))
#<class 'list'>

dir(lst)
['append',
 'clear',
 'copy',
 'count',
 'extend',
 'index',
 'insert',
 'pop',
 'remove',
 'reverse',
 'sort']

def fun():
   '''this
      is
      documentation'''
   pass

help(fun)

help(lst)
```

1. append(self, object): Append(add at the end) object to the end of the list.

```
lst = [56,98.36,None,6-9j,True,56,"Hello"]
print(lst)
```

```
#[56, 98.36, None, (6-9j), True, 56, 'Hello']
lst.append(99)
print(lst)
#[56, 98.36, None, (6-9j), True, 56, 'Hello', 99]
```

2. clear(self): Remove all items from list.

```
print(lst)
lst.clear()
print(lst)

#[56, 98.36, None, (6-9j), True, 56, 'Hello', 99]
#[]
```

clear will remove only the elements but not the list object from the memory

If we want to delete it completely from the memory we use del

```
lst = [56, 98.36, None, (6-9j), True, 56, 'Hello', 99]
del lst
print(lst)
#NameError: name 'lst' is not defined
```

3. copy(self): Return a shallow copy of the list.

We have 2 types of copy
1. Deep copy
2. Shallow copy

1. Deep copy : In Deep copy after creating a copy if we make any change on to the original or the duplicate both of them get reflected with the changes

```
lst1 = [56, 98.36, None, (6-9j), True, 56, 'Hello', 99]
lst2 = lst1
print(lst1)
print(lst2)
```

[56, 98.36, None, (6-9j), True, 56, 'Hello', 99]
[56, 98.36, None, (6-9j), True, 56, 'Hello', 99]

lst1[3] = 37
print(lst1)
print(lst2)
[56, 98.36, None, 37, True, 56, 'Hello', 99]
[56, 98.36, None, 37, True, 56, 'Hello', 99]

lst2[4] = False
print(lst1)
print(lst2)
[56, 98.36, None, 37, False, 56, 'Hello', 99]
[56, 98.36, None, 37, False, 56, 'Hello', 99]

2. Shallow copy: In Shallow copy after creating a copy if we make any change on to the original or the duplicate , the changes are limited to that object only but for not the other

lst1 = [56, 98.36, None, (6-9j), True, 56, 'Hello', 99]
lst2 = lst1.copy()
print(lst1)
print(lst2)

lst1[2]="One"
print(lst1)
print(lst2)
[56, 98.36, 'One', (6-9j), True, 56, 'Hello', 99]
[56, 98.36, None, (6-9j), True, 56, 'Hello', 99]

lst2[4]=False
print(lst1)
print(lst2)
[56, 98.36, 'One', (6-9j), True, 56, 'Hello', 99]

[56, 98.36, None, (6-9j), False, 56, 'Hello', 99]

4. count(self, value): Return number of occurrences of value.

lst1 = [56, 98.36, None, (6-9j), True, 56, 'Hello', 99]

len(lst1)
#8

lst1.count(56)
#2

5. extend(self, iterable): Extend list by appending elements from the iterable.

print(lst1)
lst1.extend([89,56,23])
print(lst1)

[56, 98.36, None, (6-9j), True, 56, 'Hello', 99, 89, 56, 23]
[56, 98.36, None, (6-9j), True, 56, 'Hello', 99, 89, 56, 23, 89, 56, 23]

print(lst1)
lst1.extend((39,26,13))
print(lst1)

[56, 98.36, None, (6-9j), True, 56, 'Hello', 99, 89, 56, 23, 89, 56, 23]
[56, 98.36, None, (6-9j), True, 56, 'Hello', 99, 89, 56, 23, 89, 56, 23, 39, 26, 13]

6. index(value, start=0, stop=9223372036854775807)
Return first index of value.
Raises ValueError if the value is not present.

print(lst1)
print(lst1.index(56))

[56, 98.36, None, (6-9j), True, 56, 'Hello', 99, 89, 56, 23, 89, 56, 23, 39, 26, 13]
#0

print(lst1.index(56,1))
#5

print(lst1.index(56,6,len(lst1)))
#9

7. insert(index, object):Insert object at the given index and move the element present in that index to the next index.

print(lst1)
#[56, 98.36, None, (6-9j), True, 56, 'Hello', 99, 89, 56, 23, 89, 56, 23, 39, 26, 13]

lst1.insert(4,236)
print(lst1)
[56, 98.36, None, (6-9j), 236, True, 56, 'Hello', 99, 89, 56, 23, 89, 56, 23, 39, 26, 13]

8. pop(index=-1): Remove and return item at index (default last).
Raises IndexError if list is empty or index is out of range.

print(lst1)
#[56, 98.36, None, (6-9j), 236, True, 56, 'Hello', 99, 89, 56, 23, 89, 56, 23, 39, 26, 13]
print(lst1.pop())
#13
print(lst1)
#[56, 98.36, None, (6-9j), 236, True, 56, 'Hello', 99, 89, 56, 23, 89, 56, 23, 39, 26]

print(lst1.pop(5))
print(lst1)
#True
#[56, 98.36, None, (6-9j), 236, 56, 'Hello', 99, 89, 56, 23, 89, 56, 23, 39, 26]

print(lst1.pop(99))

print(lst1)
#IndexError: pop index out of range

9. remove(value): Remove first occurrence of value.
Raises ValueError if the value is not present.

print(lst1)
#[56, 98.36, None, (6-9j), 236, 56, 'Hello', 99, 89, 56, 23, 89, 56, 23, 39, 26]

lst1.remove(236)
print(lst1)

#[56, 98.36, None, (6-9j), 56, 'Hello', 99, 89, 56, 23, 89, 56, 23, 39, 26]

lst1.remove(56)
print(lst1)

#[98.36, None, (6-9j), 56, 'Hello', 99, 89, 56, 23, 89, 56, 23, 39, 26]

lst1.remove(111)
#ValueError: list.remove(x): x not in list