Goeduhub Technologies - ML Training - Task 2

Registration ID: GO_STP_939

Name: Manoj Kannan D

1) Is a list mutable?

Ans : Yes, List in python is mutable

In [1]:
```
lst = ['this','is','an','example',1234]
lst[4] = 'list'
print(lst)
```

```
['this', 'is', 'an', 'example', 'list']
```

2) Does a list need to be homogeneous?

Ans: No, lists can be collection of different datatypes.

In [2]:
```
lst = ['alpha','numeric',100,0.2]
print(lst)
```

```
['alpha', 'numeric', 100, 0.2]
```

3) What is the difference between a list and a tuple.

Ans: Lists - Represented by [ ] and its mutable. Tuple - Represented by () and its immutable.

4) How to find the number of elements in the list?

Ans : The len() function returns the number of elements in a list.

In [3]:
```
lst = ['this','is','an','example',1234]
print(len(lst))
```

```
5
```

5) How to check whether the list is empty or not?

Ans: Using condition statement (if len(list) == 0), if the condition is True, the list is empty.

In [4]:
```
lst = ['this','is','an','example',1234]
if len(lst) == 0:
    print('list is empty')
else:
    print('list has elements')
```

```
list has elements
```

6) How to find the first and last element of the list?

Ans: Elements in list can be accessed using index numbers.

First element can be aaccessed using index - 0 and last element can be accessed using index - (len(list)-1) or using negative indexing.

In [5]:
```
lst = [1,2,3,4,5,6]
print('first element:',lst[0])
print('last element:',lst[len(lst)-1])
print('last element:',lst[-1])
```

```
first element: 1
last element: 6
last element: 6
```

In [ ]:
```

```

In [ ]:
```
1
```

7) How to find the largest and lowest value in the list?

Ans: Use max() and min() to find the maximum and minimum of a list.

In [6]:
```
1 lst = [1,2,3,4,5,6]
2 print('minimum element:',min(lst))
3 print('maximum element:',max(lst))
```

```
minimum element: 1
maximum element: 6
```

8) How to access elements of the list?

Ans: Elements in list can be accessed using index numbers.

In [7]:
```
1 lst = [111,222,333,444,555,666,777]
2 print('Access single element:',lst[0])
3 print('Access multiple elements(using list slicing):',lst[1:4])
```

```
Access single element: 111
Access multiple elements(using list slicing): [222, 333, 444]
```

9) Remove elements at a specific index

Ans: Using pop function, we can remove elements at a specific index

In [8]:
```
1 lst = [111,222,333,444,555,666,777]
2 lst.pop(2)
3 print(lst)
```

```
[111, 222, 444, 555, 666, 777]
```

10) Remove elements in a list between 2 indices

Ans : To remove elements in a list between 2 indices can be done using 'del' keyword.

In [9]:
```
1 lst = [111,222,333,444,555,666,777]
2 del lst[4:6]
3 print(lst)
```

```
[111, 222, 333, 444, 777]
```

11) Return every 2nd element in a list between 2 indices

In [10]:
```
1 lst = [111,222,333,444,555,666,777]
2 print(lst[1:6:2])
```

```
[222, 444, 666]
```

12) Get the first element from each nested list in a list

In [11]:
```
1 lst = [[1,2,3],[4,5,6]]
2 for row in lst:
3     print(row[0])
```

```
1
4
```

13) How to modify elements of the list?

Ans: Using assignment operator ('=')

In [12]:
```
1 lst = ['this','is','an','example',1234]
2 lst[4] = 'list'
3 print(lst)
```

```
['this', 'is', 'an', 'example', 'list']
```

In [ ]:
```
1
```

14) How to concatenate two lists?

Ans : Two lists can be concatenated using the + operator.

In [13]:
```
1 lst1 = [1,2,3,4,5,6]
2 lst2 = [111,222,333,444,555,666]
3 print(lst1 + lst2)
```

```
[1, 2, 3, 4, 5, 6, 111, 222, 333, 444, 555, 666]
```

15) How to add two lists element-wise in python?

In [14]:
```
1 lst1 = [1,2,3,4,5,6]
2 lst2 = [111,222,333,444,555,666]
3 sum_list = []
4 for i in range(len(lst1)):
5     sum_list.append(lst1[i]+lst2[i])
6 print(sum_list)
```

```
[112, 224, 336, 448, 560, 672]
```

16) Difference between del and clear?

Ans : clear - delete only the contents of a list. del - deletes the complete list from the memory

In [15]:
```
1 lst = [1,2,3,4,5,6]
2 lst.clear()
3 print('list after clearing:',lst)
4
5 lst = [1,2,3,4,5,6]
6 del lst
7 print(lst) # error: name 'lst' is not defined
```

```
list after clearing: []

---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-15-6324bbf32dfd> in <module>
      5 lst = [1,2,3,4,5,6]
      6 del lst
----> 7 print(lst) # error: name 'lst' is not defined

NameError: name 'lst' is not defined
```

17) Difference between remove and pop?

Ans : Both are used to remove elements from the list.

remove() requires the element an argument whereas pop() requires the index of the element as argument.

In [16]:
```
1 lst = [1,2,3,4,5,6]
2 lst.pop(2) # removes element at index 2
3 print(lst)
4
5 lst = [1,2,3,4,5,6]
6 lst.remove(2) # removes element 2
7 print(lst)
```

```
[1, 2, 4, 5, 6]
[1, 3, 4, 5, 6]
```

18) Difference between append and extend?

Ans: append() adds a single element to the end of the list, but extend() can add multiple elements to the end of the list.

In [17]:
```
lst = [1,2,3,4,5,6]
lst.append(7)
print(lst)

lst = [1,2,3,4,5,6]
lst.extend([7,8,9,10])
print(lst)
```

```
[1, 2, 3, 4, 5, 6, 7]
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

19) Difference between Indexing and Slicing?

Ans: Indexing is used to obtain individual elements using index number while Slicing is used to obtain a sequence of elements using corresponding index numbers.

In [18]:
```
lst = [111,222,333,444,555,666,777]
print('Access single element:',lst[0]) #Indexing
print('Access multiple elements:',lst[1:4]) #Slicing
```

```
Access single element: 111
Access multiple elements: [222, 333, 444]
```

20) Difference between sort and sorted?

Ans: sort - sorts the actual list (inplace). sorted - sorts a copy of list (not inplace)

In [19]:
```
#using sorted
lst = [4,1,7,4,9,2]
print(sorted(lst))
print('list has no changes', lst)

#using sort()
lst = [4,1,7,4,9,2]
lst.sort()
print('list is modified',lst)
```

```
[1, 2, 4, 4, 7, 9]
list has no changes [4, 1, 7, 4, 9, 2]
list is modified [1, 2, 4, 4, 7, 9]
```

21) Difference between reverse and reversed?

In [20]:
```
# using reversed
# reversed() function returns the reversed list, So it has to assigned to a variable.
# reversed() does not modify actual list.
lst = [1,2,3,4,5,6]
new = list(reversed(lst))
print(new)

# using reverse
# reverse() method does not return any value, but modifies the actual list
lst = [1,2,3,4,5,6]
lst.reverse()
print(lst)
```

```
[6, 5, 4, 3, 2, 1]
[6, 5, 4, 3, 2, 1]
```

22) Difference between copy and deepcopy?

ANS: copy() - constructs a new compound object and any changes made to a copy of object do reflect in the original object.

deepcopy() - constructs a new compound object and any changes made to a copy of object do not reflect in the original object.

In [21]:
```
import copy
act_list = [[1, 1, 1], [2, 2, 2], [3, 3, 3]]
copy_list = copy.copy(act_list)

act_list[0][0] = 'AA' # updation of elements occur in both lists

print('Actual list:', act_list)
print('Copy list', copy_list)

# both share different memory
print(id(act_list)==id(copy_list))

# but if the list elements are same, they share same memory.
# on updation of new values to list elements, then elements share different memory
print(id(act_list[0])==id(copy_list[0]))
```

```
Actual list: [['AA', 1, 1], [2, 2, 2], [3, 3, 3]]
Copy list [['AA', 1, 1], [2, 2, 2], [3, 3, 3]]
False
True
```

In [22]:
```
import copy
act_list = [[1, 1, 1], [2, 2, 2], [3, 3, 3]]
copy_list = copy.deepcopy(act_list)

act_list[0][0] = 'AA' # lists are independent, so no updation on deepcopy

print('Actual list:', act_list)
print('Copy list', copy_list)

# both share different memory
print(id(act_list)==id(copy_list))

# but if the list elements are same, they share same memory.
# on updation of new values to list elements, then elements share different memory
print(id(act_list[0])==id(copy_list[0]))
```

```
Actual list: [['AA', 1, 1], [2, 2, 2], [3, 3, 3]]
Copy list [[1, 1, 1], [2, 2, 2], [3, 3, 3]]
False
False
```

23) How to remove duplicate elements in the list?

Ans : Duplicate items can be removed in the list by using set() function.

In [23]:
```
lst = [1,1,4,5,6,2,2,2,3,4,5,6]
new = list(set(lst))
print('list without duplicates:',new)
```

```
list without duplicates: [1, 2, 3, 4, 5, 6]
```

24) How to find an index of an element in the python list?

Ans : index() function is used to find an index of an element in lists.

In [24]:
```
lst = [1,10,22,33,2,3,66,99,4,5,6]
print('index of element 3 is',lst.index(3))
```

```
index of element 3 is 5
```

25) How to find the occurrences of an element in the python list?

Ans : count() function can be used to find the number of occurrences of an element in the list.

In [25]:
```
lst = [1,1,4,5,6,2,2,2,3,4,5,6]
print('No. of occurances of element 2 is', lst.count(2))
```

```
No. of occurances of element 2 is 3
```

26) How to insert an item at a given position?

Ans : insert() function is used to insert an item at a given position.

Syntax: list.insert(index,element)

In [26]:
```
lst = [1,2,3,4,5,6]
lst.insert(0,999)
print(lst)
```

```
[999, 1, 2, 3, 4, 5, 6]
```

27) How to check if an item is in the list?

Ans: Conditional statement if can be used

In [27]:
```
lst = [1,2,3,4,5,6]
if 5 in lst:
    print('5 is present in the List')
else:
    print('5 is not present in the List')
```

```
5 is present in the List
```

28) How to flatten a list in python?

In [28]:
```
lst = [[1, 1, 1], [2, 2, 2], [3, 3, 3]]
flatten = []
for row in lst:
    for i in row:
        flatten.append(i)
print(flatten)
```

```
[1, 1, 1, 2, 2, 2, 3, 3, 3]
```

29) How to convert python list to other data structures like set, tuple, dictionary

In [29]:
```
lst = [1,2,3,4,5,6]
lst_set = set(lst)
print(type(lst_set))

lst_tup = tuple(lst)
print(type(lst_tup))

lst_dict = {'list_values' : lst}
print(type(lst_dict))
```

```
<class 'set'>
<class 'tuple'>
<class 'dict'>
```

30) How to apply a function to all items in the list?

Ans: Using map() function

In [30]:
```
lst = [1,2,3,4,5,6]
new = list(map(lambda x:x**2,lst)) # create new list with square of each element in lst
print(lst)
print(new)
```

```
[1, 2, 3, 4, 5, 6]
[1, 4, 9, 16, 25, 36]
```

31) How to filter the elements based on a function in a python list?

Ans: Using filter() function

In [31]:
```
lst = [1,2,3,4,5,6]
new = list(filter(lambda x: x>2,lst)) # returns new list for elements >2 in lst
print(lst)
print(new)
```

```
[1, 2, 3, 4, 5, 6]
[3, 4, 5, 6]
```

32) How python lists are stored in memory?

Ans: The list object consists of two internal parts; one for object header (first element), and one separately allocated array of object references.