Task2

GO_STP_7364

Assignment/Task 2

-

Question 1 : Is a list mutable?

Mutable means modifying the content and state after the sequence is created. Since, we can append,insert,pop etc on list therefore, List is mutuable.

In [6]:
```
#LET L be a list
L=[1,2,3,"hello","world","5.7"]
print(L)

#changing the 2nd elt of the list
L[3]="new"
print(L)

#removing elt from the list
L.remove(3)
L
```

```
[1, 2, 3, 'hello', 'world', '5.7']
[1, 2, 3, 'new', 'world', '5.7']
```

Out[6]:
```
[1, 2, 'new', 'world', '5.7']
```

Question 2 : Does a list need to be homogeneous?

In [8]:
```
# No, the item in list can be homogenous or hetrogenous. It totally depends upon

A=["1",1.,"one",1] #this list contains hetrogenous items in it.
A
```

Out[8]:
```
['1', 1.0, 'one', 1]
```

Question 3 : What is the difference between a list and a tuple ?

In [80]:
```
# List: Mutable
# Tuple:Immutable

list=[1,2,3,4,"hi","you"]
tuple=(1,2,3,4,"hi","you")

print(list,tuple,sep="\n")

#checking mutability
list.append("Bye")
#tuple.append("Bye") #error would be returned.
```

```
[1, 2, 3, 4, 'hi', 'you']
(1, 2, 3, 4, 'hi', 'you')
```

Question 4: How to find the number of elements in the list?

In [15]:
```
# Length function is used to find the length of elts in list

list =[1,2,3,4,5,6]
len(list)
```

Out[15]: 6

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

In [14]:
```
l=[]
if len(l)==0:
    print("The list is empty")

lis=[1,23,4]
if lis==0:
    print("The list is empty.")
else:
    print("The list is not empty.")
```

```
The list is empty
The list is not empty.
```

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

In [2]:
```
# 1st element can be accessed by list[0] and last elt by list[-1]

list=[1,2,3,7,8,5,"Jyoti","Bisht"]
print(list[0],list[-1],sep="\n")
```

```
1
Bisht
```

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

In [20]:
```
list=[9,6,5,66,70]
print(max(list),min(list),sep="\n")

# or sort the list and find the 1st and last element

list.sort()
print("min" ,list[0],"and max" , list[-1])
```

```
70
5
min 5 and max 70
```

Question 8 : How to access elements of the list?

In [7]:
```
# Indexing allows us to access single element from the list while slicing allows

print(list[4])
print(list[2:])
```

```
70
[5, 66, 70]
```

Question 9 :Remove elements in a list before a specific index

In [15]:
```
# It can be done by - pop(). remove() removes an element by value and pop removes
list.pop()
list
```

Out[15]:
```
[9, 6, 66]
```

Question 10 : Remove elements in a list between 2 indices

In [5]:
```
# By del we can remove elementa in a list between 2 indices

list1=[1,3,5,7,2,3,4,88]
del list1[3:5]
list1
```

Out[5]:
```
[1, 3, 5, 3, 4, 88]
```

Question 11: Return every 2nd element in a list between 2 indices

In [6]:
```
# By slicing and striding we can do that
list2=[2,4,1,3,5,6,3,2]
list2[::2]
list2
```

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

Question 12 : Get the first element from each nested list in a list

In [12]:
```
list=[[1,2,3],[4,2],[3,1,2,34,5],[6,77,89,9,7,3]]
l=[x[0] for x in list]
l
```

Out[12]:
```
[1, 4, 3, 6]
```

Question 13: How to modify elements of the list?

In [13]:
```
# modification can be done by list[index]=newvalue

list=[1,3,5,7,83,3]
list[4]=1000
list
```

Out[13]:
```
[1, 3, 5, 7, 1000, 3]
```

Question 14 : How to concatenate two lists?

In [53]:
```
l1=["apple","mango","orange"]
l2=[1,2,3,4]

# first method : append
l1.append(l2)
print(l1)

#Second method: add
l1=["apple","mango","orange"]
l2=[1,2,3,4]
l3=l1+l2
print(l3)

#third : list comprehension
l1=["apple","mango","orange"]
l2=[1,2,3,4]
a=[j for i in (l1,l2) for j in i]
print(a)

#fourth : extend
l1=["apple","mango","orange"]
l2=[1,2,3,4]
l1.extend(l2)
print(l1)

#fifth : * operator

l1=["apple","mango","orange"]
l2=[1,2,3,4]
l3=[*l1,*l2]
print(l3)
```

```
['apple', 'mango', 'orange', [1, 2, 3, 4]]
['apple', 'mango', 'orange', 1, 2, 3, 4]
['apple', 'mango', 'orange', 1, 2, 3, 4]
['apple', 'mango', 'orange', 1, 2, 3, 4]
['apple', 'mango', 'orange', 1, 2, 3, 4]
```

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

In [79]:
```
# method 1 : list comprehension

l1=[1,2,3,4,5]
l2=[8,4,7,9,8]

l3=[i+j for i,j in zip(l1,l2)]
print(l3)

#method 2 : map

#l4=list(map(lambda(i,j):i+j , zip(l1,l2)))


#l4
res = map(lambda i,j: i + j, zip(l1,l2))
```

```
[9, 6, 10, 13, 13]
```

Question 16 : Difference between del and clear?

In [73]:
```
# on applying del the whole sequence will be deleted but on applying clear it wil
l1=[1,2,3,4,6,78,4]
l1.clear()
print(l1)

l2=[1,2,3,4,5]
del l2
#l2 doesn't exist now.
```

```
[]
```

Question 17 :Difference between remove and pop?

In [88]:
```
# Remove :  syntax - remove(elt)
#removes that certain element present in the argument ,if multiples elements matc

list1=[2,3,3,6,7]
list1.remove(3)
print(list1)

#pop : syntax - pop() / pop(index)
#if no argument is passed it will remove last elt and will pop last elt in output

list2=[2,3,3,6,7]
list2.pop(2)
```

```
[2, 3, 6, 7]
```

Out[88]:
```
3
```

Question 18 :Difference between append and extend?

In [91]:
```
# Appends takes a single argument and adds that argument at the end of the list w

list=[1,2,3,4,5]
list.append([1,2,3,4])
print(list)

#extend Iterates over its argument and adding each element to the list and extend

list1=[1,2,3,4]
list1.extend([3,5,6])
print(list1)
```

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

Question 19 :Difference between indexing and Slicing?

In [93]:
```
#indexing is used to obtain single element while slicing can retrive multiple ite

l1=[1,2,3,5,6,7]
print(l1[2])

print(l1[2:4])
```

```
3
[3, 5]
```

Question 20 :Difference between sort and sorted?

In [3]:
```
# sort - syntax: list.sort()
#sorts the  sequence of the element in the list in ascending order if no argument
# sorted- syntax : sorted(list)
#creates a new sorted list in which sequence of the items in list is in descendir


l1=[1.2,5,7,4,33,6]
l2=sorted(l1)
l1.sort()

print(l1,l2,sep="\n")
```

```
[1.2, 4, 5, 6, 7, 33]
[1.2, 4, 5, 6, 7, 33]
```

Question 21 : Difference between reverse and reversed?

In [15]:
```
# reverse function inverts the order of items in the list and modifies the origir
# Reversed function returns an iterator that access the given sequence in the rev

list=[1,2,4,"jyoti","bisht"]
list.reverse()
print(list)

my_list = [1, 2, 4, 3, 5]

for i in reversed(my_list):
    print(i)
```

```
['bisht', 'jyoti', 4, 2, 1]
5
3
4
2
1
```

Question 22 :Difference between copy and deepcopy?

copy : In Python, we use = operator to create a copy of an object. You may think that this creates a new object; it doesn't. It only creates a new variable that shares the reference of the original object. Even their Ids are same. So, if you want to modify any values in new_list or old_list, the change is visible in both. Essentially, sometimes you may want to have the original values unchanged and only modify the new values or vice versa. In Python, there are two ways to create copies:

1.Shallow Copy 2.Deep Copy

To make these copy work, we use the copy module. A deep copy creates a new object .This means, both the old_list and the new_list are independent and changes made in old_list will not affect new_list.

In [21]:
```
a=[1,2,3,4,4,5]
b=a #copying items of a to b
a[3]=9
print(a,b)

#deepcopy
import copy
x=[2,4,5,6,7,33,3]
c=copy.deepcopy(x)
x[-1]=0
print(x,c) #acting independent, no change is reflected in c.
```

```
[1, 2, 3, 9, 4, 5] [1, 2, 3, 9, 4, 5]
[2, 4, 5, 6, 7, 33, 0] [2, 4, 5, 6, 7, 33, 3]
```

Question 23:How to remove duplicate elements in the list?

In [39]:
```
# we can convert the items of list into dictionary keys (Since keys are unique) c
# or convert the list into set

a=[1,2,3,4,4,3,2,1,2]
b=set(a)
print(b)
```

```
{1, 2, 3, 4}
```

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

In [40]:
```
# it returns the index of specific element in the list. It takes maximum 3 argume

list=["a","b","c","d","e",2,4,5]
list.index("c",1,5)
```

Out[40]: 2

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

In [41]:
```
# Count is used for that purpose.
a=[2,3,5,5,6]
a.count(5)
```

Out[41]: 2

Question 26 :How to insert an item at a given position?

In [42]:
```
a=[2,3,4,2,3,"a","c"]
a.insert(0,"b")
a
```

Out[42]:
```
['b', 2, 3, 4, 2, 3, 'a', 'c']
```

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

In [43]:
```
# in membership checks that.

a=["s","d",2,3,4,4]
if 2 in a:
    print("Yes,it is in list")
else:
    print("Not in list")
```

```
Yes,it is in list
```

Question 28 :How to flatten a list in python?

In [47]:
```
# 1.using list comprehension
a=[[1,2,3],[3,4,5],[5,6,7]]

b=[j for i in a for j in i]
print(b)

# loop
c=[]
for i in a:
    c+=(i)
print(c)
```

```
[1, 2, 3, 3, 4, 5, 5, 6, 7]
[1, 2, 3, 3, 4, 5, 5, 6, 7]
```

Question 29 :How to convert python list to other data structures like set, tuple, dictionary?

In [85]:
```
list1=[1,2,3,4,5,3,2,4]

set1=set(list1)
print(set1)

tuple1=tuple(list1)
print(tuple1)
```

```
{1, 2, 3, 4, 5}
(1, 2, 3, 4, 5, 3, 2, 4)
```

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

1 . map function 2. list comprehension 3. lambda function

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

In [80]:
```
# build-in function filter() does that.

# syntax : filter(function,iterable)

x=[1,2,3,4,5]
a=filter(lambda i : i>2,x)
```

Question 32: How python lists are stored in memory?

Lists are stored in distinct chunks of memory which are linked together with pointers, which enables efficient use of memory generally and doesn't require resizing