Rules and Guidelines of List Data Structure

- A list is an ordered collection of data.
- Lists are defined using square brackets [].
- Lists are indexable, meaning you can access elements using their positions.
- Lists are **mutable,** allowing you to modify or remove items within them.
- You can create a list using the list() function or by simply using square brackets [].
- Items in a list are separated by commas.
- **Note:** Lists can hold any type of data.

In [1]:
```
print("List Functions: ", [i for i in dir(list) if "__" not in i])
```

```
List Functions:  ['append', 'clear', 'copy', 'count', 'extend', 'index', 'insert', 'pop', 'remove', 'reverse', 'sort']
```

List Creation

In [2]:
```
lst = []
```

In [3]:
```
lst
```

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

In [4]:
```
type(lst)
```

Out[4]:
```
list
```

In [5]:
```
newLst = list()
```

In [6]:
```
newLst
```

Out[6]:
```
[]
```

In [7]:
```
lst = [1,2,3,4,5]
```

In [8]:
```
lst
```

Out[8]:
```
[1, 2, 3, 4, 5]
```

In [9]:
```
lst = [1,2,3,6,4,"Aman","Elaina","Eric",10.24,True,False,["A","C"]]
```

In [10]:
```
print(lst)
```

```
[1, 2, 3, 6, 4, 'Aman', 'Elaina', 'Eric', 10.24, True, False, ['A', 'C']]
```

List Append

Using this function we can **append an item at the end of list.** Item can be anything.

In [11]:
```
lst = [1,5,7,"A"]
```

In [12]:
```
lst.append("Pushpanjali")
```

In [13]:
```
lst
```

Out[13]:
```
[1, 5, 7, 'A', 'Pushpanjali']
```

In [14]:
```
newlst = ["Modi","Biden","Jocinda"]
```

In [15]:
```
lst.append(newlst)
```

In [16]:
```
print(lst)
```

```
[1, 5, 7, 'A', 'Pushpanjali', ['Modi', 'Biden', 'Jocinda']]
```

List Extend

This function allows us to extend an existing list by appending elements from another iterable (such as a list, tuple, or string). Each item from the iterable is extracted and added individually to the end of the original list.

In [17]:
```
lst = [1,5,7,"A"]
```

In [18]:
```
newlst = ["Modi","Biden","Jocinda"]
```

In [19]:
```
lst.extend(newlst)
```

In [20]:
```
print(lst)
```

```
[1, 5, 7, 'A', 'Modi', 'Biden', 'Jocinda']
```

Can you add 100 using the extend function?

In [21]:
```
lst.extend([100])
```

In [22]:
```
print(lst)
```

```
[1, 5, 7, 'A', 'Modi', 'Biden', 'Jocinda', 100]
```

In [23]:
```
lst.extend(100)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[23], line 1
----> 1 lst.extend(100)

TypeError: 'int' object is not iterable
```

In [26]:
```
lst
```

Out[26]:
```
[1, 5, 7, 'A', 'Modi', 'Biden', 'Jocinda', 100]
```

In [27]:
```
lst.extend("Modi")
```

In [28]:
```
print(lst)
```

```
[1, 5, 7, 'A', 'Modi', 'Biden', 'Jocinda', 100, 'M', 'o', 'd', 'i']
```

In [29]:
```
[100]
```

Out[29]:
```
[100]
```

In [30]:
```
list("Manisha")
```

Out[30]:
```
['M', 'a', 'n', 'i', 's', 'h', 'a']
```

In [31]:
```
name = "Abhishek"
```

In [32]:
```
list(name)
```

Out[32]:
```
['A', 'b', 'h', 'i', 's', 'h', 'e', 'k']
```

Index

In [38]:
```
lst = [10,12,34,56,78,34,22,34,56,78,34,56,78,34,12,34,56,67,23,56,34]
```

In [39]:
```
lst.index(12)
```

Out[39]:
```
1
```

In [40]:
```
lst.index(78)
```

Out[40]:
```
4
```

In [41]:
```
lst.index(34)
```

Out[41]:
```
2
```

In [42]:
```
lst.index(34,2+1)
```

Out[42]:
```
5
```

In [43]:
```
lst.index(34,lst.index(34)+1)
```

Out[43]:
```
5
```

In [45]:
```
print(lst)
```

```
[10, 12, 34, 56, 78, 34, 22, 34, 56, 78, 34, 56, 78, 34, 12, 34, 56, 67, 23, 56, 34]
```

In [46]:
```
lst.index(34, lst.index(34,lst.index(34)+1)+1)
```

Out[46]: 7

In [47]:
```
lst.index(34,lst.index(34, lst.index(34,lst.index(34)+1)+1)+1)
```

Out[47]: 10

In [48]:
```
lst[10]
```

Out[48]: 34

Insert

Based on the index we can insert at item.

In [50]:
```
newlst
```

Out[50]: ['Modi', 'Biden', 'Jocinda']

In [52]:
```
newlst.insert(2,"Debiprasad")
```

In [53]:
```
newlst
```

Out[53]: ['Modi', 'Biden', 'Debiprasad', 'Jocinda']

In [54]:
```
newlst.append("Bider")
```

In [55]:
```
newlst
```

Out[55]: ['Modi', 'Biden', 'Debiprasad', 'Jocinda', 'Bider']

In [56]:
```
newlst.index("Biden")
```

Out[56]: 1

In [57]:
```
newlst.insert(2,"Abhishek")
```

In [58]:
```
newlst
```

Out[58]: ['Modi', 'Biden', 'Abhishek', 'Debiprasad', 'Jocinda', 'Bider']

In [59]:
```
print(lst)
```

```
[10, 12, 34, 56, 78, 34, 22, 34, 56, 78, 34, 56, 78, 34, 12, 34, 56, 67, 23, 56, 34]
```

In [64]:
```
lst.index(78, lst.index(78) + 1)
```

Out[64]: 9

In [65]:
```
lst.index(78, lst.index(78, lst.index(78) + 1)+1)
```

Out[65]: 12

In [67]:
```
lst.insert(lst.index(78, lst.index(78, lst.index(78) + 1)+1),"Abhinav")
```

In [68]:
```
print(lst)
```

```
[10, 12, 34, 56, 78, 34, 22, 34, 56, 78, 34, 56, 'Abhinav', 78, 34, 12, 34, 56, 67, 23, 56, 34]
```

In [69]:
```
lst.insert(lst.index(78, lst.index(78, lst.index(78) + 1)+1) + 1,
           "Abhinav2")
```

In [70]:
```
print(lst)
```

```
[10, 12, 34, 56, 78, 34, 22, 34, 56, 78, 34, 56, 'Abhinav', 78, 'Abhinav2', 34, 12, 34, 56, 67, 23, 56, 34]
```

Clear Method

In [71]:
```
lst.clear()
```

In [72]:
```
lst
```

Out[72]:
```
[]
```

List count

this function will tell you the occurence of an item.

In [73]:
```
lst = [10,12,34,56,78,34,12,34,56,67,23,56,34]
```

In [74]:
```
lst
```

Out[74]:
```
[10, 12, 34, 56, 78, 34, 12, 34, 56, 67, 23, 56, 34]
```

In [75]:
```
lst.count(34)
```

Out[75]:
```
4
```

Pop, Remove, Del

pop

The pop function is used to remove an item from a list based on its index. It also returns the removed item. If no index is provided, the function removes and returns the last item in the list.

If the list is empty, attempting to use the pop function will raise an error.

In [76]:
```
lst
```

Out[76]:
```
[10, 12, 34, 56, 78, 34, 12, 34, 56, 67, 23, 56, 34]
```

In [77]:
```
lst.pop()
```

Out[77]: 34

In [78]:
```
lst
```

Out[78]: [10, 12, 34, 56, 78, 34, 12, 34, 56, 67, 23, 56]

In [79]:
```
lst.pop(4)
```

Out[79]: 78

In [80]:
```
lst
```

Out[80]: [10, 12, 34, 56, 34, 12, 34, 56, 67, 23, 56]

Remove

Using this function we can remove an item based on item's name, this function does not return anything like pop function.

In [83]:
```
lst
```

Out[83]: [10, 12, 34, 56, 34, 12, 34, 56, 67, 23, 56]

In [82]:
```
lst.remove("Abhishek")
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[82], line 1
----> 1 lst.remove("Abhishek")

ValueError: list.remove(x): x not in list
```

In [84]:
```
lst.remove(34)
```

In [85]:
```
lst
```

Out[85]: [10, 12, 56, 34, 12, 34, 56, 67, 23, 56]

In [86]:
```
lst.remove(23)
```

In [87]:
```
lst
```

Out[87]:
```
[10, 12, 56, 34, 12, 34, 56, 67, 56]
```

Del

Del is not the function of list, it is keyword, using del keyword we can delete entire list or specific item from the list but based on the index.

Del does not return anything like remove.

In [88]:
```
lst
```

Out[88]:
```
[10, 12, 56, 34, 12, 34, 56, 67, 56]
```

In [89]:
```
del lst[-1]
```

In [90]:
```
lst
```

Out[90]:
```
[10, 12, 56, 34, 12, 34, 56, 67]
```

In [91]:
```
del lst[3]
```

In [92]:
```
lst
```

Out[92]:
```
[10, 12, 56, 12, 34, 56, 67]
```

In [93]:
```
del lst
```

In [95]:
```
lst
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[95], line 1
----> 1 lst

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

In [97]:
```
newlst
```

Out[97]:
```
['Modi', 'Biden', 'Abhishek', 'Debiprasad', 'Jocinda', 'Bider']
```

In [98]:
```
deleteDataBase = []
```

In [99]:
```
deleteDataBase.append(newlst.pop(newlst.index("Abhishek")))
```

In [100…
```
newlst
```

Out[100…
```
['Modi', 'Biden', 'Debiprasad', 'Jocinda', 'Bider']
```

In [101…
```
"Abhishek" in deleteDataBase
```

Out[101…
```
True
```

In [102…
```
deleteDataBase
```

Out[102…
```
['Abhishek']
```

In [103…
```
data = [12,3,4,56,100,34,"AB","BD","MD","BCA"]
```

In [104…
```
data
```

Out[104…
```
[12, 3, 4, 56, 100, 34, 'AB', 'BD', 'MD', 'BCA']
```

In [105…
```
data.reverse()
```

In [106…
```
data
```

Out[106… ['BCA', 'MD', 'BD', 'AB', 34, 100, 56, 4, 3, 12]

In [108…
```
newdata = [34,56,10,2,40,-45,23,45,70]
```

In [109…
```
newdata.sort()
```

In [110…
```
newdata
```

Out[110… [-45, 2, 10, 23, 34, 40, 45, 56, 70]

In [111…
```
newdata = [34,56,10,2,40,-45,23,45,70]
```

In [112…
```
newdata.sort(reverse=True)
```

In [113…
```
newdata
```

Out[113… [70, 56, 45, 40, 34, 23, 10, 2, -45]

In [114…
```
copy_newdata = newdata.copy()
```

In [115…
```
copy_newdata
```

Out[115… [70, 56, 45, 40, 34, 23, 10, 2, -45]

In [116…
```
newdata
```

Out[116… [70, 56, 45, 40, 34, 23, 10, 2, -45]

In [117…
```
y = newdata
```

In [118…
```
y
```

Out[118… [70, 56, 45, 40, 34, 23, 10, 2, -45]

In [119…
```
newdata
```

Out[119… [70, 56, 45, 40, 34, 23, 10, 2, -45]

In [122…
```
lst = ["I","love","Python","Programming"]
```

In [123…
```
lst
```

Out[123… ['I', 'love', 'Python', 'Programming']

In [124…
```
"".join(lst)
```

Out[124… 'IlovePythonProgramming'

In [125…
```
" ".join(lst)
```

Out[125… 'I love Python Programming'

In [126…
```
name = "AdityaGarg"
```

In [127…
```
"*".join(name)
```

Out[127… 'A*d*i*t*y*a*G*a*r*g'

In [128…
```
"->".join("Abhinav")
```

Out[128… 'A->b->h->i->n->a->v'

In [129…
```
lst
```

Out[129… ['I', 'love', 'Python', 'Programming']

In [130…
```
" | ".join(lst)
```

Out[130… 'I | love | Python | Programming'

In [131…
```
lst
```

Out[131… ['I', 'love', 'Python', 'Programming']

In [132…
```
lst[0]
```

Out[132…
```
'I'
```

In [133…
```
lst[0] = "Aman"
```

In [134…
```
lst
```

Out[134…
```
['Aman', 'love', 'Python', 'Programming']
```

In [135…
```
lst = [1,2,3,4,(22,33,44),0,10]
```

In [139…
```
lst[-3][-1] = "Aditya"
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[139], line 1
----> 1 lst[-3][-1] = "Aditya"

TypeError: 'tuple' object does not support item assignment
```

In [138…
```
lst[-3]
```

Out[138…
```
(22, 33, 44)
```

Assignment

In [147…
```
lst = [1,2,3,44,55,6, [34,56,[34,56,[[56,"Z",8,9,[34,56,"X"]]]]]]
```

Can you change X and Z by their ascii char?

In [151…
```
lst = [1,2,3,44,55,6, [34,56,[34,56,[[56,"Z",8,9,[34,56,78]]]]]]
```

Can you sort only the list : 34,56,78 in desc order

In [ ]: