Assignment-1

10 write how to use various list function in python

List:-

→ Lists are used to store multiple items in a single variable.

→ List's are one bulit-in data types in Python used to store collection of data, the other 3 are TUPle, set and Dictionary, all with different qualities and usage

→ List are created using square brackets.

example:-

```
Fruits =[ "apple", "banana", "cherry"]
print (Fruits)
```

output will be " apple, banana, cherry"

→ Print statement will give the output which is Fruits.

List items:-

→ List items are ordered, changeable, and allow duplicate values

→ List items are indexed, the first item has index[0], the second item has index[1]

Access List Items:-

→ List items are indexed and you can access them by referring to the index number.

example:-

```
Fruits = [ "apple", "banana", "cherry"]

Print(Fruits[1])
```

→ The output will be banana because it access the first position "1" position is banana.

change Item value

→ To change the value of a specific item, refer to index number.

example:-

```
Fruits = ["apple", "banana", "cherry"]
Fruits[1] = "blackcurrant"
```

→ In this the Fruit in position "1" banana change with fruit blackcurrant.

output = apple, blackcurrent, cherry

Add List Items:-

→ To add an item to the end of the list, use the append() method:

example:-

```
Fruits = ["apple", "banana", "cherry"]
Fruits.append("orange")
```

Here append is used to add fruit orange with the remaining fruits.

output = ["apple, banana, cherry, orange]

Remove List item.

→ The remove() method removes the specified item

→ remove the Data (or) delete the Data

→ used to delete in list.

example:-

```
Fruits = ["apple", "banana", "cherry"]

Fruits.remove("banana")
print(Fruits)
```

Here the fruit banana in position 1 will be removed from the fruits.

output = "apple, cherry"]

→ If there are more than one item with the specified value, the remove() method removes the first occurance.

Copy List:

→ you cannot copy a list simply by typing list2 = list1, because: list2 will only be a reference to list1, and changes made in list1 will automatically also be made in list2

→ There are ways to make a copy, one way is to use the built-in List method copy()

example:-

```
Fruits = ["apple", "banana", "cherry"]
mylist = Fruits.copy()
print(mylist)
```

Here mylist = Fruits.copy() "copy()" help to copy fruits and store in mylist.

output = "apple, banana, cherry"

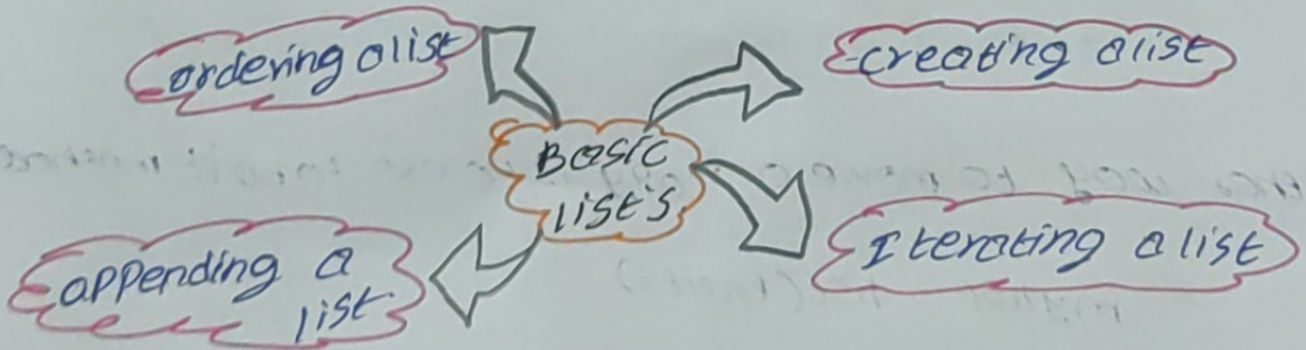
→ Another way to make a copy is to use to built method list()

```
mylist = list(fruits)
```

List methods

Method	Description
append()	→ Adds an element at the end of list
clear()	→ removes all the element from list.
copy()	→ Return a copy of the list
count()	→ Return the number of element with the specified value.
extend()	→ Add the elements to a list, to end of current list
index()	→ Return the index of first element with specified value.
insert()	→ Add an element at the specified position
pop().	→ Removes the element with the specified position
remove()	→ remove the item with specified value
reverse()	→ reverse the order of the list.
sort()	→ sort the list.

- Each method has it seperate use.
- This are the main and all list in the python
- They are four Basic list in python



ordering a list

→ when we say that a list are ordered, it mean that the item have defined order, and the order will not change.

→ If you add new item to a list, then the new item will be placed at the end of the list.

example:-

```
a = [1, 2, "RAM", 3.50, "Rahul", 5, 6]
b = [1, 2, 5, "RAM", 3.50, "Rahul", 6]
a == b
```

OUTPUT = False.

→ The indistinguishable components we remembered for the two record; however, the subsequent rundown changed the file position of the fifth component,

→ which is against the rundown's planned request. False is returned when the two lists are compared.

correct code:-

```
a = [1, 2, "RAM", 3.50, "Rahul", 5, 6]
b = [1, 2, "RAM", 3.50, "Rahul", 5, 6]
a = b
```

OUTPUT:- True.

→ Records forever protect the component structure Because of this, it is arranged collection of thing.

→ It can arrange the order the element

→ specified list

→ easy to understand.

Iterating a list

→ we can use a for loop to iterate over the elements of a list

example:-

```
languages = ["telugu", "tamil", "Hindhi"]

#iterating through the list
for language in languages:
    print(language)
```

we can use the in keyword to check if an item exists in the list or not- from example.

```
languages =["telugu","tamil","Hindhi"]

print("english" in lanuages)  #False
print("telugu" in lanuages)   #True.
```

⇒ here the languages iterating into the language

```
output:- telugu
         tamil
         Hindhi
```

⇒ english is not present in languages. "english" in languages evaluates to false

⇒ "telugu" is present in languages. "telugu" in languages evaluate to true.

Creating a list:

→ we create a list by placing elements inside [ ] separated by commas.

example:

```
age = [19, 26, 23]
print (age)
```

→ Here, we have created a list named ages with 3 integer items

→ print will be used to show (or) to get the output.

output will be [19, 26, 23]

→ a list can store elements of different types (integer, float, string, etc)

→ store duplicate element.

```
# list with different element
list1 = [1, "Hello", 3.4]
# list with duplicate element.
list1 = [1, "Hello", 3.4, "Hello", 1]

# empty list
list2 = [ ]
```

→ we can also create a list using the list() construction

∴ "list()"

→ In python, lists are address (or) ordered and each item in a list is associated with a number. The number is know as a list index.

appending a list

→ The append() method adds an item to the end of the list

→ appending method using to add the element in the list

example:

```
currencies = ["dollor", 'Euro', 'pound']
#appending "yen" to the list
currencies.append('yen')
print(currencies)
```

→ Here append is used to add the element yen in the currencies.

→ The yen is added at the at the last

∴ output = [dollor, Euro, pound, yen]

→ The syntax is append() (or) list.append(item)

→ The method takes a single argument

→ An item like (number, string, list etc.) to be added at the end of the list.

→ The method doesn't return any value(returns None)

() another example :

```
animals = ["cat", "dog", "rat"]
animals.append("pig")
print(animals)
```

output: [cat, dog, rat, pig]

→ Here using append() function we add animal pig in the animals

Slicing of a List

→ In python, it is possible to access a portion of a list using the slicing operator: For example

```
#list slicing in python
mylist = ["p", "x", "o", "g", "r", "a", "m"]
#items from index 2 to index 4
print(mylist[2:5])
```

output will be [o, g, r]

→ Here mylist[2:5] returns a list with items from index 2 to index 4

→ When we slice list, the start index is inclusive, but the end index is exclusive.

change a list

→ python lists are mutable meaning lists are changeable and we can change items of a list by assigning new values using = operator.

example

```
language = ['python', 'swift', 'c++']
languages[2] = 'c'
print (languages)
```

Here the languages in position "2" swift will change with 'c'

- print will show the output.

output = [python, c, c++]

Length List

→ we can use the len() function to find the size of a list

example:-

```
languages = ['python', 'swift', 'c++']
print("List:", languages)
print("Total:", len(languages))
```

→ Here len() will calculate the length of the languages. [python, swift, c++]

output = Total element 3