#### EXPERIMENT NO : 2A

**NAME : AKASH RAMKRIT YADAV**

**ID.NO: VU4F2122016**

**BATCH : A**

**BRANCH : IT**

**DIV : A**

##### Python programs to implement Different List.

**Aim:- Python programs to implement Different List.**

###### THEORY:

OUTPUT:

Python 3.11.0a4 (main, Jan 17 2022, 12:57:32) [MSC v.1929 32 bit

(Intel)] on win32

Type "help", "copyright", "credits" or "license()" for more information.

#AKASH RAMKRIT YADAV #ID NO:VU4F2122016 DATE:25/01/2023

#Create a List:

Thislist = ["mango", "banana", "cherry"] print(thislist)

['mango', 'banana', 'cherry']

1. # Creating a List of numbers

list=[1,2,3,4,5,6,7,8] print("\n LIST OF NUMBER IS:")

LIST OF NUMBER IS:

print(list)

[1, 2, 3, 4, 5, 6, 7, 8]

[1, 2, 3, 4, 5, 6, 7, 8]

[1, 2, 3, 4, 5, 6, 7, 8]

1. # Creating a List of strings and accessing

*# using index*

list = ["AKASH","RAMKRIT","YADAV"]

print("\n LIST ITEM ARE:")

LIST ITEM ARE:

print(list[0]) AKASH

print(list[1]) RAMKRIT

print(list[2])

YADAV

print(list[-1]) YADAV

print( list[0],list[1],list[2]) AKASH RAMKRIT YADAV

1. # Creating a List with # the use of Numbers

***# (Having duplicate values)***

a=[1,2,2,3,4,4,5,6,6,7,7,8]

print("\nList with the use of Numbers: ")

List with the use of Numbers:

print(a)

[1, 2, 2, 3, 4, 4, 5, 6, 6, 7, 7, 8]

1. # Creating a List with # mixed type of values

***# (Having numbers and strings)***

list=[1,2,3,"akash","have","good",4,"Day"] print("\nList with the use of Mixed Values: ",list)

List with the use of Mixed Values: [1, 2, 3, 'akash', 'have', 'good', 4, 'Day']

1. # Creating a Multi-Dimensional List # (By Nesting a list inside a List)

akash=[['HAVE','A'],['GOOD','DAY']]

# accessing an element from the # Multi-Dimensional List using # index number

print(akash[0][1]) A

print("\n Accessing a element from a Multi-Dimensional list:\n",akash[0][0])

Accessing a element from a Multi-Dimensional list: HAVE

print("\n Accessing a element from a Multi-Dimensional list:\n",akash[0][0],akash[0][1],akash[1][0],akash[1][1])

Accessing a element from a Multi-Dimensional list: HAVE A GOOD DAY

1. # negative indexing

***# accessing an element using negative indexing***

l=[1,5,7,'AKASH',6,'YADAV','ONE']

# print the last element of list print('last element of list is :\n',l[-1]) last element of list is :

ONE

print(' 2nd last element of list is :\n',l[-2]) 2nd last element of list is :

YADAV

print('4th last element of list is :\n',l[-4]) 4th last element of list is :

AKASH

print(' 5th last element of list is :\n',l[-5]) 5th last element of list is :

7

1. #Python len()

***#Python len() is used to get the length of the list.***

# Creating a List l=[]

print(len(l)) 0

# Creating a List of numbers l1=[1,2,3,4,5]

print(len(l1)) 5

# Creating a List of alphabets l2=['akash','yadav','ram']

###### #ACCESS LIST ITEMS:

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

*# Creating a List of strings and accessing # using index*

***#positive Indexing***

list = ["AKASH","RAMKRIT","YADAV"]

print("\n LIST ITEM ARE:",list)

LIST ITEM ARE: ['AKASH', 'RAMKRIT', 'YADAV']

print(list[0]) AKASH

print(list[1]) RAMKRIT

print(list[2]) YADAV

#Negative Indexing

Negative indexing means start from the end

-1 refers to the last item, -2 refers to the second last item etc.

list = ["AKASH","RAMKRIT","YADAV"]

print("\n LIST ITEM ARE:",list)

LIST ITEM ARE: ['AKASH', 'RAMKRIT', 'YADAV']

print(list[-1]) YADAV

print(list[-2]) RAMKRIT

print(list[-3]) AKASH

print( list[0],list[1],list[2]) AKASH RAMKRIT YADAV

# Range of Indexes

You can specify a range of indexes by specifying where to start and where to end the range.

When specifying a range, the return value will be a new list with the specified items.

**Creating list:**

**#common list for all below example:**

***list = ["AKASH","RAMKRIT","YADAV","1",2,3,4,5,6,"viram","ram","shyam"]***

print(len(list)) 12

print(list[0:6])

['AKASH', 'RAMKRIT', 'YADAV', '1', 2, 3]

print(list[0:3])

['AKASH', 'RAMKRIT', 'YADAV']

print(list[3:3]) []

print(list[3:13])

['1', 2, 3, 4, 5, 6, 'viram', 'ram', 'shyam']

***# By leaving out the start value, the range will start at the first item:***

*list = ["AKASH","RAMKRIT","YADAV","1",2,3,4,5,6,"viram","ram","shyam"]*

*print(list[:4])*

*['AKASH', 'RAMKRIT', 'YADAV', '1'****]***

***# By leaving out the end value, the range will go on to the end of the list:***

*list = ["AKASH","RAMKRIT","YADAV","1",2,3,4,5,6,"viram","ram","shyam"]*

***print(list[3:])***

***['1', 2, 3, 4, 5, 6, 'viram', 'ram', 'shyam']***

###### # Range of Negative Indexes

Specify negative indexes if you want to start the search from the end of the list:

*list = ["AKASH","RAMKRIT","YADAV","1",2,3,4,5,6,"viram","ram","shyam"]*

*print(list[-12:-1])*

*['AKASH', 'RAMKRIT', 'YADAV', '1', 2, 3, 4, 5, 6, 'viram', 'ram']*

***#*Check if Item Exists**

To determine if a specified item is present in a list use the in keyword:

*list = ["AKASH","RAMKRIT","YADAV","1",2,3,4,5,6,"viram","ram","shyam"]*

***if "AKASH" in list:***

***print("Yes, 'AKASH' is in the NAME list")***

***Yes, 'AKASH' is in the NAME list***

## *#* Change List Items

1. Change Item Value

To change the value of a specific item, refer to the index number: #Change the second item:

**list = ["AKASH","RAMKRIT","YADAV","1",2,3,4,5,6,"viram","ram","shyam"]**

***list[1]="NILESH"***

print(list)

['AKASH', 'NILESH', 'YADAV', '1', 2, 3, 4, 5, 6, 'viram', 'ram', 'shyam']

1. **Change a Range of Item Values**

To change the value of items within a specific range, define a list with the new values, and refer to the range of index numbers where you want to insert the new values:

***#Change the values "ram" and "shyam" with the values "raju" and "ravi":***

**list = ["AKASH","RAMKRIT","YADAV","1",2,3,4,5,6,"viram","ram","shyam"]**

***list[10:12]=["raju" ,"ravi"] print(list)***

***['AKASH', 'NILESH', 'YADAV', '1', 2, 3, 4, 5, 6, 'viram', 'raju', 'ravi']***

**# If you insert *less* items than you replace, the new items will be inserted where you specified, and the remaining items will move accordingly**:

***Change the 2nd and 12th value by replacing it with* one *value:***

***List=['AKASH', 'NILESH', 'YADAV', '1', 2, 3, 4, 5, 6, 'viram', 'raju',***

***'ravi']***

***list[1:12]=["KRISHNA"]***

***print(list) ['AKASH', 'KRISHNA']***

**#Add List Items**

1. **Insert Items**

To insert a new list item, without replacing any of the existing values, we can use the insert() method.

The insert() method inserts an item at the specified index:

***list = ["AKASH","RAMKRIT","YADAV","1",2,3,4,5,6,"viram","ram","shyam"]***

***list.insert(2,"RADHA-KRISHNA") print(list)***

***['AKASH', 'RAMKRIT', 'RADHA-KRISHNA', 'YADAV', '1', 2, 3, 4, 5, 6, 'viram',***

***'ram', 'shyam']***

1. **Append Items**

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

***list = ["AKASH","RAMKRIT","YADAV","1",2,3,4,5,6,"viram","ram","shyam"]***

***list.append("NARENDRA MODI") print(list)***

***['AKASH', 'RAMKRIT', 'YADAV', '1', 2, 3, 4, 5, 6, 'viram', 'ram', 'shyam', 'NARENDRA MODI']***

### Extend List / Add Any Iterable

To append elements from *another list* to the current list, use the extend() method.

The extend() method does not have to append *lists*, you can add any iterable object (tuples, sets, dictionaries etc.).

A1=["AKASH","YADAV"]

A2=[1,2,3,4,5,"RAM"]

A1.extend(A2) print(A1)

['AKASH', 'YADAV', 1, 2, 3, 4, 5, 'RAM']

# #Remove List Items

## Remove Specified Item

The remove() method removes the specified item.

***Remove ”RAMKRIT” [2nd element] from list list = ["AKASH","RAMKRIT","YADAV"]***

***list.remove("RAMKRIT") print(list)***

***['AKASH', 'YADAV']***

## Remove Specified Index

1. ***The pop() method removes the specified index***.

Remove the second item:

***list = ["AKASH","RAMKRIT","YADAV"]***

***list.pop(1) 'RAMKRIT'***

***print(list)***

***['AKASH', 'YADAV']***

1. **If you do not specify the index, the pop() method removes the last item.**

***list = ["AKASH","RAMKRIT","YADAV"]***

***list.pop() 'YADAV'***

***print(list) ['AKASH', 'RAMKRIT']***

1. **The del keyword also removes the specified index:**

Remove the first item:

*list = ["AKASH","RAMKRIT","YADAV"]*

*del list[1] print(list) ['AKASH', 'YADAV']*

1. The del keyword can also delete the list completely.

list = ["AKASH","RAMKRIT","YADAV"]

del list print(list)

<class 'list'**>**

1. **Clear the List**

The clear() method empties the list.

The list still remains, but it has no content.

**Clear the list content:**

list = ["AKASH","RAMKRIT","YADAV"]

list.clear()

print(list)

[]

# #Loop Lists

### Loop Through a List

You can loop through the list items by using a for loop:

**Print all items in the list, one by one:**

*list = ["AKASH","RAMKRIT","YADAV"]*

*for x in list: print(x)*

*AKASH RAMKRIT YADAV*

### Looping Using List Comprehension

List Comprehension offers the shortest syntax for looping through lists:

*list = ["AKASH","RAMKRIT","YADAV"]*

*[print(x) for x in list]*

*AKASH RAMKRIT YADAV*

*[None, None, None]*

# #Sort Lists

## Sort List Alphanumerically

List objects have a sort() method that will sort the list alphanumerically, ascending, by default:

*list=["AKASH","RAMKRIT","YADAV","BOB","SUNY","VIRAM","SURAJ","KAVIN","NARENDR A","NILESH"]*

*list.sort() print(list)*

*['AKASH', 'BOB', 'KAVIN', 'NARENDRA', 'NILESH', 'RAMKRIT', 'SUNY', 'SURAJ', 'VIRAM', 'YADAV']*

1. **Sort the list numerically:**

*list=[1,2,4,6,7,8,34,56,78,23,9,34,56,78,76,92]*

*list.sort() print(list)*

*[1, 2, 4, 6, 7, 8, 9, 23, 34, 34, 56, 56, 76, 78, 78, 92]*

#### Sort Descending

To sort descending, use the keyword argument reverse = True:

*list=["AKASH","RAMKRIT","YADAV","BOB","SUNY","VIRAM","SURAJ","KAV IN","NARENDRA","NILESH"]*

*list.sort(reverse = True) print(list)*

*['YADAV', 'VIRAM', 'SURAJ', 'SUNY', 'RAMKRIT', 'NILESH', 'NARENDRA', 'KAVIN', 'BOB', 'AKASH']*

1. **Sort the list descending**

*list=[2,4,6,3,1,7,8,45,67,65,23,56,89,98,543,12,455,675]*

*list.sort(reverse=True) print(list)*

*[675, 543, 455, 98, 89, 67, 65, 56, 45, 23, 12, 8, 7, 6, 4, 3, 2, 1]*

# #Copy Lists

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.

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

*list = ["AKASH","RAMKRIT","YADAV"]*

*AKASH=list.copy() print(AKASH)*

*['AKASH', 'RAMKRIT', 'YADAV']*

1. ***Another way to make a copy is to use the built-in method list().***

l1 = ["akash", "ramkrit ", "yadav"] mylist = list(l1)

print(mylist) [‘akash’,’ramkrit ‘, ‘yadav’]