**Program:**

l1=['books','periodicals','newspapers'] # creating the list

''' Add elements to the list '''

l1.append('manuscripts') # adding elements at end of the list

l1.insert(2,'maps') # adding elements at the specific position

l1.extend(['films','prints']) # adding more number of elements

print(l1)

''' Changing elements of the list '''

l1[5]='ebooks' # changing 'films' to 'ebooks'

l1[:3]=['books','maps','periodicals'] # changing sequence of elements

print(l1)

''' Accessing the list elements by index values '''

print(l1[0]) # positive indexing

print(l1[-2]) # negative indexing

print(l1[3][:3]) # nested indexing

print(l1[::]) # accessing all the elements

print(l1[::-1]) # accessign all elements in reverse

''' Concatenation '''

print(l1+['films']) # concatenating two lsits

print(l1+['films']+['maps']) # concatenating more than two lists

''' Repetation '''

print(l1[0]\*3) # repetation of the single element

print(['books','maps','ebooks']\*2) # repetation of the list

''' Check with membership operator '''

print('books' in l1) # return True

print('books' not in l1) # return False

''' Operations with build-in functions '''

print(len(l1)) # lenght of list

print(l1.count('books')) # return the count of particular element

print(l1.index('newspapers')) # return the index value

print(l1.sort()) # this function sort the list

print(sorted('books')) # this does not effect the original list

l1.reverse() # reverse the list

print(l1[::-1]) # reverse the list using slicing

''' Slicing '''

print(l1[:3]) # slicing the elements between 0 and 3

print(l1[::2])

print(l1[1:5:2])

print(l1)

''' Deleting '''

l1.pop(0) # removing the element at index 0

print(l1)

l1.remove('manuscripts') # remove the element 'films'

print(l1)

del l1[2]# remove element at index 0 with del keyword

print(l1)

l1.clear() # clears all element in the list

print(l1)

del l1 # deletes the list l1

print(l1)

**Output:**

['books', 'periodicals', 'maps', 'newspapers', 'manuscripts', 'films', 'prints']

['books', 'maps', 'periodicals', 'newspapers', 'manuscripts', 'ebooks', 'prints']

books

ebooks

new

['books', 'maps', 'periodicals', 'newspapers', 'manuscripts', 'ebooks', 'prints']

['prints', 'ebooks', 'manuscripts', 'newspapers', 'periodicals', 'maps', 'books']

['books', 'maps', 'periodicals', 'newspapers', 'manuscripts', 'ebooks', 'prints', 'films']

['books', 'maps', 'periodicals', 'newspapers', 'manuscripts', 'ebooks', 'prints', 'films', 'maps']

booksbooksbooks

['books', 'maps', 'ebooks', 'books', 'maps', 'ebooks']

True

False

7

1

3

None

['b', 'k', 'o', 'o', 's']

['books', 'ebooks', 'manuscripts', 'maps', 'newspapers', 'periodicals', 'prints']

['prints', 'periodicals', 'newspapers']

['prints', 'newspapers', 'manuscripts', 'books']

['periodicals', 'maps']

['prints', 'periodicals', 'newspapers', 'maps', 'manuscripts', 'ebooks', 'books']

['periodicals', 'newspapers', 'maps', 'manuscripts', 'ebooks', 'books']

['periodicals', 'newspapers', 'maps', 'ebooks', 'books']

['periodicals', 'newspapers', 'ebooks', 'books']

[]

**--------------------------------------------------------------**

**NameError**Traceback (most recent call last)

Input **In [16]**, in <cell line: 58>**()**

56 print(l1)

57 **del**l1 # deletes the list l1

**---> 58**print(l1)

**NameError**: name 'l1' is not defined

**Program:**

''' Creating tuple '''

tup1=('Ferrari 296','plug\_in\_hybrid',2022) # creating with parenthesis

tup2='Sports car','twin-turbocharged\_v6' # creating without parenthesis

print(tup1)

''' Packing and Unpacking '''

model,ctype,year=tup1 # Unpacking the tuple

tup1=model,ctype,year # Packing the tuple

print(tup1)

''' Concatenating and repeating '''

tup1=tup1+tup2 # concatenation

print(tup1)

print(tup2\*2) # repeatation

''' Accessing the elements '''

print(tup1[0]) # positive indexing

print(tup1[-1]) # negative indexing

fori in tup1: # traversing the tuple

print(i,end='')

''' Slicing '''

print(tup1[:3])

print(tup1[2:])

print(tup1[::-1])

''' Build-in functions ''’

print(len(tup1)) # find the length of tuple

print(tuple(tup1[0]))

print(type(tup1))

''' Deleting '''

del tup1 # to delete the tuple

print(tup1)

**Output:**

('Ferrari 296', 'plug\_in\_hybrid', 2022)

('Ferrari 296', 'plug\_in\_hybrid', 2022)

('Ferrari 296', 'plug\_in\_hybrid', 2022, 'Sports car', 'twin-turbocharged\_v6')

('Sports car', 'twin-turbocharged\_v6', 'Sports car', 'twin-turbocharged\_v6')

Ferrari 296

twin-turbocharged\_v6

Ferrari 296plug\_in\_hybrid2022Sports cartwin-turbocharged\_v6('Ferrari 296', 'plug\_in\_hybrid', 2022)

(2022, 'Sports car', 'twin-turbocharged\_v6')

('twin-turbocharged\_v6', 'Sports car', 2022, 'plug\_in\_hybrid', 'Ferrari 296')

5

('F', 'e', 'r', 'r', 'a', 'r', 'i', ' ', '2', '9', '6')

<class 'tuple'>

**---------------------------------------------------------------------------**

**NameError**Traceback (most recent call last)

Input **In [12]**, in <cell line: 34>**()**

32 ''' Deleting '''

33 **del** tup1 # to delete the tuple

**---> 34**print(tup1)

**NameError**: name 'tup1' is not defined

**Program:**

s1={1,2,3} # creating a set

''' Adding elements to set '''

s1.add(4) # adding single element

s1.update([1,2,5]) # adding multiple elements

s1.update([6,7],{2,5}) # addign list and set

print(s1)

''' Accessing elements '''

fori in s1: # iterate the set

print(i,end=' ')

''' Build-in functions '''

print(len(s1)) # returns the length of set

print(type(1)) # returns the datatype

print(max(s1)) # maximum value

print(min(s1)) # minimum value

print(sum(s1)) # sum

''' Deleting '''

s1.pop() # removing last element

s1.discard(7) # removing 7

print(s1)

s1.remove(6) # removing 6

print(s1)

s1.clear() # removing all

print(s1)

del s1

print(s1)

**Output:**

{1, 2, 3, 4, 5, 6, 7}

1 2 3 4 5 6 7 7

<class 'int'>

7

1

28

{2, 3, 4, 5, 6}

{2, 3, 4, 5}

set()

---------------------------------------------------------------------------

NameErrorTraceback (most recent call last)

Input In [35], in <cell line: 30>()

28 print(s1)

29 del s1

---> 30 print(s1)

NameError: name 's1' is not defined