18/10/22

Ch-7 List in Python

A list is a collection of values. or an ordered sequence of values which can be a string, integer, float, even a list also.

⇒ It is separated by commas and ~~out~~ enclosed within a square bracket.

⇒ It allows duplicate values and elements of the list, can be accessed by its index.

⇒ List are mutable i.e. values of list can be modified which means it will not create a new list to make new changes.

e.g → L = ["Raj", 01, 20, 30, 40.2]

There are 3 types of list :-

① Empty list

② Long list

③ Nested list

1) Empty list

L = [ ]

2) Long list

l = [1, 2, ...]

3) Nested list

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

Creating a List by using the function :- list()

Example :- L = list ("computer")
print (L)

o/p ⇒ ['c', 'o', 'm', 'p', 'u', 't', 'e', 'r']

18/10/22

Practise Questions :- (Strings in Python)

Q1) WAP to count the frequency of a character in a string.

Ans- str = input ("Enter any string")
ch = input ("Enter character")
print (str.count(ch))

Q2) WAP to accept a string and return a string having first letter of each word in uppercase / capital.

Ans- str = input ("Enter any string")
print (str.title())

Q3) WAP to accept a string & display the following:-

i) No. of uppercase characters.
ii) No. of lowercase characters.
iii) Total no. of alphabets.
iv) No. of digits.

Ans-
```
str = input("Enter any string")
u = 0
L = 0
d = 0
l = len(str)
for i in range(l):
    if str[i].isupper():
        u = u + 1
    if str[i].islower():
        L = L + 1
    if str[i].isdigit():
        d = d + 1
print("Total Upper Case Characters are:", u)
print("Total Lowercase Characters are:", L)
print("Total characters are:", L + u)
print("Total digits are:", d)
```

Q4) Write a program to reverse a string :-

Ans-
```
str = input("Enter any string")
print(str[::-1])
```

Q5) Fill in the blanks :-

i) The data or text enclosed with single quote, double quote or triple quote is known as String.

ii) The string which is having 0 characters is known as Empty String.

iii) Each character in a string has a unique position or ID in the text, that is known as index.

iv) The index of string starts from 0 to length-1 in forward direction.

v) The process of accessing a string character by character is known as Traversing.

vi) The * (Replication) ~~oper~~ operator is used to repeat the word or specified text n times.

vii) To join more than 2 words you can use + (Concatenation) operator.

viii) The **in** operator returns True if a character or specified String is available in the given string.

ix) The **in** and **not in** operators are membership operators.

x) To reverse a string using string slice, **s[::-1]** is an easy way to do so.

Q6) Multiple Choice Questions :-

I) Which of the following is the correct way of indexes of strings, begin from the reverse in backward direction?

i) 0 to length - 1
iii) - length to -1
ii) -1 to - length ✓
iv) length - 1 to 0 ✓

Ans → (ii) & (iv).

II) Iterating through the various elements of a string, one character at a time is called _____.

Ans = (b) String Traversing.

III) When you write S[-5] for a five letter word, python will return

Ans- (a) S[0].

IV) What will be the result of this code:
('2' + 3)

Ans- (a) Error

V) What will be the result of this code:
'*' * 3

Ans- (c) ***

VI) The expression "CSIP" in "TutorialAICSIP" returns

Ans- (b) True

VII) The expression "Tutor" <= "TutorialAICSIP" returns.

Ans- (b) True.

VIII) Which of the following function returns the ASCII code for specified character?

Ans- (c) ord()

IX) Which of the following function returns the character from given integer code ?.

Ans- (d) chr()

X) What will be the result of: s['len(s), -3]

Ans- (b) Error.

XI) To print first four letters from the string, which of the option(s) is/are correct?

i) s[:3]
ii) s[0:4]
iii) s[:4]
iv) s[0:3]

Ans = Option (ii) & (iii).

XII) Which of the following is not a correct string operation in python?

a) 'Tut' + 'or'
b) 'Tutor' * 2
c) 'Tutor' + 2
d) 2 * 'Tutor'

Ans- (c) 'Tutor' + 2.

20/10/22

Indexing :-

A list consist of any collection of values stored acc. to its index. It can contain (+ve) index or (-ve) index.

	0	1	2	3	4	5	...
List =	10,	20,	30,	'A',	'B',	40	]
...	-6	-5	-4	-3	-2	-1	

List [3] = 'A'
List [-3] = 'A'
List [20] = Error (Index Out of Range)

Traversing a List :-

- By using 'in' operator inside for loop.

```
List= [10,20,30]
for i in list:
        print (i)
```

O/P 10
20
30

- Using range function.

```
List=['P','y','t','h','o','N']

n= len(List)
for i in range(n):
    print(list[i])
```

O/P ⇒

```
P
y
t
h
o
N
```

- Using while loop:

```
list = [10,20,30]
i = 0
n= len(list)
while i<=n:
    print(list[i])
    i+=1.
```

List are mutable data type which means, values can be changed within the list only.

```
>> list = [10, 20, 30]
   list[0] = 50
   print (list)
O/P => [50, 20, 30]
```

Comparing the values in a List :-

```
>>> l = [10, 20, 30]
>>> l1 = [10, 20, 30]
>>> l3 = [10, [10, 20], 30]
>>> l1 == l      → True
>>> l1 == l3     → False
```

Comparison	Result	Reason
[1,2,3,4] < [4,5,6]	True	1<4
[1,2,3,4] < [1,5,2,3]	True	2<5
[1,2,3,4] < [1,2,3,2]	False	4>2

26/10/22

Operations on List

1) Concatenation
2) Repetation
3) Membership testing
4) Indexing
5) Slicing

1) Concatenation → Concatenation or joining is a process in which multiple list can be combined together with the help of (+) operator.

e.g. i)
```
l1 = [1,2,3,4]
l2 = [5,6,7,8]
l3 = l1 + l2
print (l3)
```

O/P ⇒ [1,2,3,4,5,6,7,8]

ii)
```
l1 = [1,2,3,4]
l2 = l1 + 5
print (l2)
```

O/P ⇒ Error.

2) Repetition → This (*) operator replicates the list for a specified number of times and creates a new list.

e.g. → l1 = [1,2,3,4]
print (l1 * 2)
O/P ⇒ [1,2,3,4,1,2,3,4]

3) Membership Testing → Membership testing is an operation carried out to check or test whether a particular element is a member of that sequence or not. 'in' and 'not in' is used.

e.g. → ① l1 = [1,2,3,4]
print (40 in l1)
O/P ⇒ False

② l1 = [1,2,3,4]
print (40 not in l1)
O/P ⇒ True.

4) Indexing → List supports +ve as well as -ve indexing which starts from 0 or -1 respectively.

0 1 2 3 4 → +ve indexing
l1 = [0, 1, 2, 3, 4]
-5 -4 -3 -2 -1
-ve indexing

5) Slicing → Slicing is an operation in which we can slice a particular range from a sequence

```
x = ['c', 'o', 'm', 'p', 'u', 't', 'e', 'r']
→ print (x[1:4])
⇒ ['o', 'm', 'p']
→ print (x[1:6:2])
⇒ ['o', 'p', 't']

→ print (x[3:])
⇒ ['p', 'u', 't', 'e', 'r']

→ print (x[:-5])
⇒ ['c', 'o', 'm']

→ print (x[-1:])
⇒ ['r']

→ print (x[:-2])
⇒ ['c', 'o', 'm', 'p', 'u', 't']
```

Nested List :-

→ When a list appear as an element of the another list, is called a Nested list. A list can have more than 1 list inside it.

e.g. → l = [1, 2, 3, [4, 5, 6], 7]

```
          0  1  2     3     4
l = [ 1, 2, 3, [4, 5, 6], 7 ]
```

→ print (l[3])
⇒ [4, 5, 6]

→ print (l[3][1])
⇒ 4

Copying a List :-

There are 3 ways to copy a list into another list.

Method I :-

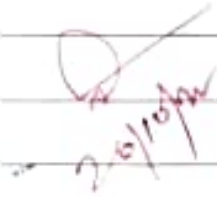
```
l = [1, 2, 3, 4]
l1 = l[:]
print (l1)
```

Method II :-

```
l = [1, 2, 3, 4]
l1 = list [l]
print (l1)
```

Method III :-

```
l = [1, 2, 3, 4]
l1 = l.copy ()
print (l1)
```



28/10/22

Built in Functions

Python offers various functions that can alter the element of the list. The various functions are :-

1) append() → This function adds a single element at the end of the list. It doesn't create a new list, rather it modifies the original list.

Syntax

- list.append (item)

e.g.→ l = [10, 20, 30]
l.append (40)
print (l)
o/p ⇒ [10, 20, 30, 40]

2) ~~ent~~ extend() → This method ~~at~~ add one list at the end of another list. All the item of the list are added at the end of the already created list.

Syntax ~~list.append~~
list.extend (list1)

e.g. → l = [10, 20, 30]
l1 = [40, 50]
l.extend (l1)
print (l)

O/P → [10, 20, 30, 40, 50]

3) insert ( ) → This function can be used to insert an element at a specified index. This function takes two argument :-
i) Index Number & ii) Value.

Syntax list.insert (index, value)

e.g. → l = [10, 20, 30]
l.insert (2, 25)
print (l)

O/P → [10, 20, 25, 30]

4) reverse() → This function reverse the order of the element in a list. It does not creates a new list, it changes the value in a place of an item from the existing list.

Syntax
list.reverse()

e.g.→ l = [10,20,40] o/p ⇒ [40,20,10]
l.reverse()

5) index() → This function returns the index of first matched item from the list. If the item is present in the list, it will return the index value, otherwise it will display index_value_error.

Syntax
list.index(value)

e.g.→ list = [10,20,30,30] o/p ⇒ 2.
list.index(30)

6) len() ⇒ This function returns the length of the given list.

Syntax
len(list)

e.g.→ list = [10, 20, 30, 30] o/p ⇒ 4.
print (len(list))

7) sort() → This function sorts the item of the list in ascending order. The modification is done in the existing list. It does not create a new list.

Syntax
list.sort()

e.g.→ list = [Blue, Green, 20, 30]
list.sort()
print (list)

o/p.) [20, 30, Blue, Green]

8) count() → This function counts how many times the given element occurs.

Syntax → list.count(element)

e.g.→ l = [10, 20, 30, 20]
print(l.count(20)) ⇒ 2
print(l.count(50)) ⇒ 0

9) clear() → This method removes all the values present in the list. It doesn't take any parameters only returns the empty list.

Syntax → list.clear()

e.g.→ l = [10, 20, 30] o/p ⇒ [ ]
l.clear()
print(l)

01/11/22

Deletion Operation

Python provides operator for deleting or removing ~~not~~ an item from the list.
The methods are :-

1) Pop() → It removes the element from the specified index & also returns the removed element.
2) Del() → It removes the specified element from the list but doesn't return the deleted value.
3) remove() → The function is used when we know the specified element that is to be deleted, not the index of the element.

① l1 = [10, 20, 30]
l1.pop(2)
>>> 30

② l1 = [10, 20, 30]
l1.del(2)
>>> [10, 20]

③ l1 = [10, 20, 30]
l1.remove(20)
>>> [10, 30]

10) Max() → It returns the maximum element from the list.

```
>>> l1 = [10, 30, 100]
>>> max(l1)
>>> 100

>>> l1 = ['a', 'b', 'A']
>>> max(l1)
>>> b
```

11) min() → It returns the minimum value from the list.

```
>>> l1 = [10, 30, 100]
>>> min(l1)
>>> 10

>>> l1 = ["Rajesh", "Suman", "Ravi"]
>>> min(l1)
>>> Rajesh.
```

Q) WAP to input 10 values in a list and display the maximum and minimum value in it.

⇒
```
l = [ ]
for i in range (0, 11):
        v= int (input ("Enter value = "))
        l.append (v)
print ("Maximum value=", Max (l))
print ("Minimum value =", Min (l))
```

Q) WAP to create a list of your own choice and multiply all the even number with 10 and odd nos. with 5. Also display the updated list.

⇒
```
l = []
n = int (input ("Enter no. of values = "))
for i in range (0, n+1)
        num = int (input ("Enter value = "))
        l.append (num)
for i in range (len (l)):
        if l[i] % 2 == 0:
                l[i] = l[i] * 10
        else:
                l[i] = l[i] * 5
print ("updated list", l)
```

Q) WAP to exchange 1st half element of the list to the last half element, assuming the list have even nos. of elements

⇒
```
l= [10,20,30,40,50,60,70]
n= int(len(l)/2)
for i in range(n):
    l[i], l[n+i] = l[n+i], l[i]
print(l).
```

OR

⇒
```
l = [10,20,30,40,50,60,70]
l1 = []
l2 = []
ln = int(len(l)/2):
for i in range(ln):
    l1.append(l[i])
for j in range(ln, len(l)):
    l2.append(l[j])
l2.extend(l1)
print(l2)
```

Q) WAP to display those strings which are starting with the character 'A' or 'a' from the given list.

⇒ L= ["ANKUR", "TARUN", "SUMAN", "AKHTAR"]

```
for i in L:
    if i[0] == A or i[0] == a :
        print (i).
```

OR

```
for i in L:
    if i[0] in ('aA'):
        print (i)
```

Q) WAP to display the sum of these values which are ending with three. from the given list :-

```
L = [33, 32, 32, 63, 83]
c = 0
for i in L:
    if (i % 10 == 3):
        c = c + i
print (c).
```

Q) WAP to copy all the values from the list to the another list which are divisible by 7.

⇒

```
L = [14, 21, 63, 36, 42, 85]
M = [ ]
for i in L:
    if (i % 7 == 0):
        M.append (i)
print (M)
```

Menu-driven program to do various list operations:-

```
list1 = [22, 4, 16, 38, 13]
choice = 0
while True:
    print ("The list has following elements", list1)
    print ("\n LIST OPERATIONS")
    print ("1. Append an element")
    print ("2. Insert an element at desired position")
    print ("3. Append a list to a given list")
    print ("4. Modify an existing element")
    print ("5. Delete an existing element by its position")
    print ("6. Delete an existing element by its value")
    print ("7. Sort the list in ascending order")
    print ("8. Sort the list in descending order")
    print ("9. Display the list")
    print ("10. Exit")

    choice = int(input ("Enter choice [1-10]: "))
    if choice == 1:
        element = int (input ("Enter element to be
                  appended : "))
        list1.append (element)
        print ("The element has been appended \n")

    elif choice == 2:
        element = int (input ("Enter element "))
        pos = int (input ("Enter position of element"))
```

```
        list1.insert(pos, element)
        print("The element has been inserted\n")

    elif choice == 3:
        newList = int(input("Enter the list:"))
        list1.extend(newList1)
        print("The list has been appended\n")

    elif choice == 4:
        i = int(input("Enter the position of
        to be modified element:"))
        if i < len(myList)
                newElement = int(input("Enter element"))
                oldElement = list1[i]
                list1[i] = newElement
                print("The element has been
                        modified\n")
        else:
                print("Position of element is more than
                        length of list")

    elif choice == 5:
        i = int(input("Enter the position of the
                element to be deleted="))
        if i < len(list1):
                element = list1.pop(i)
                print("The element has been deleted
                        \n")
```

```
        else:
            print ("\n The position of element
                is exceeding list length")

    elif choice == 6:
        element = int(input("Enter the element to
                be deleted "))
        if element in list1:
            list1.remove(element)
            print ("\n The element has been deleted")
        else:
            print("\n Element is not present in list")

    elif choice == 7:
        list1.sort()
        print ("\n The list has been sorted")

    elif choice == 8:
        list1.sort(reverse = True)
        print ("\n The list has been sorted
            in reverse order")

    elif choice == 9:
        print ("\n The list is:", list1)

    elif choice == 10:
        break
```

```
else:
    print("Choice is not valid")
    print("\n\n Press any key to
                continue.....")
    ch= input()
```

40) Write a menu-driven program to do the various list operations:

a) Sort list in ascending order using bubble sort.

b) Sort list in descending order using Insertion sort.

c) Search an element.

d) Count an element.

e) Display list.

```
=> l1 = []
n = int(input("Enter no. of elements to be entered"))
print ("Enter elements")
for i in range (n):
        ele = int (input())
        l1. append (ele)
print("The entered list is: ", l1, '\n')

while True:
    print ("List Operations")
    print("1. Search an element")
    print ("2. Count an element")
    print ("3. Display list")
    print ("4. Exit \n")
    choice = int(input ("Enter choice [1-4]:"))

    if choice == 1:
        ele = int (input("Enter element"))
        for i in range (n):
            if l1 [i] == ele:
                print ("Element found at", i)
                break
            else:
                print ("Element not found")
```

```
elif choice == 2:
    ele = int(input("Enter the element"))
    freq = l1.count(ele)
    print("The frequency of",ele,"is",freq)

elif choice == 3:
    print("The list is=", l1)

elif choice == 4:
    break

else:
    print("Invalid choice")
```