STRING IN PYTHON

Definition: String is a collection of characters. Strings can be created by enclosing characters inside a single quote or double-quotes. Even triple quotes can be used in Python but generally used to represent multiline strings. Python does not have a character data type, a single character is simply a string with a length of 1.

Basics of String:

- Strings are immutable means that the contents of the string cannot be changed after it is created. At the same memory address, the new value cannot be stored. Python does not allow the programmer to change a character in a string.
  Example:
  >>>str='jaipur'
  >>>str[0]='J'
  TypeError: 'str' object does not support item assignment
  As shown in the above ex

  ample, str has the value "jaipur". An attempt to replace 'j' in the string by "J" displays a TypeError.
- Each character has its index or can be accessed using its index.
- String in python has two-way index for each location. (0, 1, 2, ……. In the forward direction and -1, -2, -3, …….. in the backward direction.)
  Example:

0	1	2	3	4	5	6
T	E	A	C	H	E	R
-7	-6	-5	-4	-3	-2	-1

- The index of string in forward direction starts from 0 and in backward direction starts from -1.
- The size of string is total number of characters present in the string. (If there are n characters in the string, then last index in forward direction would be n-1 and last index in backward direction would be –n.)
- String are stored each character in contiguous location.
- The character assignment is not supported in string because strings are immutable.

Accessing Characters in a String

As we know, string is a collection of characters and individual character can be accessed by its position called index. Square brackets can be used to access elements of the string.

```
Example:
>>>s="TEACHER"
>>>s[1]
'E'  # returns index 1 position
>>>s[-4]
'C' #returns index -4 position
```

Traversing a String: Access all elements of string, one character at a time.

s=”TEACHER” for ch in s : print(ch)	s=”TEACHER” for i in range(len(s)): print(s[i])

>>>len(s) # returns length of string.

String Operators:

A). **String concatenation Operator:** Concatenation means to join two values. In Python, + symbol is used to concatenate the strings.

```
>>>name="Jay"
>>>msg="Hello "
>>>print(msg+name)
```

'Hello Jay' #concatenated string

Note: You cannot concate numbers and strings as operands with + operator.

Example:

>>>7+’4’ # unsupported operand type(s) for +: 'int' and 'str'

It is invalid and generates an error.

B). **String repetition Operator**: It is also known as String replication operator. Replication can be performed by using * operator between the string. It will repeat the string n times, where n is the integer provided

```
>>>s="Ha"
>>> s*3
'HaHaHa'          #Replication
```

Note:You cannot have strings as n=both the operands with * operator.

Example:

>>>”Ha” * “Ha” # can't multiply sequence by non-int of type 'str'

It is invalid and generates an error.

C). Membership Operators: In and not in are two membership operators to find the appearance of a substring inside the string.**in –** Returns **True** if a character or a substring exists in the given string; otherwise, **False**

not in - Returns **True** if a character or a substring does not exist in the given string; otherwise, **False**

Example:

```
>>> "T" in "TEACHER"
True
>>> "ea" in "TEACHER "
False
>>>"CH" not in "TEACHER "
False
```

D). Comparison Operators: These operators compare two strings character by character according to their ASCII value. ASCII Values can be finding out by given functions.

Characters	ASCII (Ordinal) Value
‘0’ to ‘9’	48 to 57
‘A’ to ‘Z’	65 to 90
‘a’ to ‘z’	97 to 122

Function	Description
ord(<character>)	Returns ordinal value of a character
chr(<value>)	Returns the corresponding character

Example:

```
>>> 'abc'>'abcD'
False
>>> 'ABC'<'abc'
True
>>> 'abcd'>'aBcD'
True
>>> 'aBcD'<='abCd'
True
>>> ord('b')
98
>>> chr(65)
'A'
```

Slicing in Strings: Extracting a subpart from a main string is called slicing .It is done by using a range of indices.

Syntax:

>>>string-name[start:stop:step]

Note: it will return a string from the index **start** to stop-1.

Example:

```
>>> s="TEACHER"
```

0	1	2	3	4	5	6
T	E	A	C	H	E	R
-7	-6	-5	-4	-3	-2	-1

```
>>> s[2:6:1]
'ACHE'
>>> s[6:1:-1]
'REHCA'
>>> s[0:10:2]
'TAHR'
>>> s[-8:-3:1]
'TEAC
>>> s[  : 6 : 1]  # Missing index at start is considered as 0.
'TEACHE'
>>> s[2 :  :2] # Missing index at stop  is considered as last index.
'AHR'
>>> s[3:6: ] # Missing index at step  is considered as 1.
'CHE'
>>> s[ : :-1]
'REHCAET'
>>> s[2: :]+s[ :2 :]
'ACHERTE'
>>> s[1: 5:-1]
‘‘
```

Built-in functions of string:

str=”data structure” s1= “hello365” s2= “python” s3 = ‘4567’
s4 = ‘ ‘ s5= ‘comp34%@’

S. No.	Function	Description	Example
1	len( )	Returns the length of a string	>>>print(len(str)) 14
2	capitalize( )	Returns the copy of the string with its first character capitalized and the rest of the letters are in lowercased.	>>> s1.capitalize() 'Hello365'
3	find(sub,start,end)	Returns the index of the first occurence of a substring in the given string (case-sensitive). If the substring is not found it returns -1.	>>>s2.find("thon",1,7) 3 >>> str.find("ruct",8,13) -1
4	isalnum( )	Returns True if all characters in the string are alphanumeric (either alphabets or numbers). If not, it returns False.	>>>s1.isalnum( ) True >>>s2.isalnum( ) True >>>s3.isalnum( ) True >>>s4.isalnum( ) False >>>s5.isalnum( ) False
5	isalpha( )	Returns **True** if all characters in the string arealphabetic. **False** otherwise.	>>>s1.isalpha( ) False >>>s2.isalpha( ) True >>>s3.isalpha( ) False >>>s4.isalpha( ) False >>>s5.isalpha( ) False
6	isdigit( )	Returns **True** if all the characters in the string aredigits. **False** otherwise.	>>>s1.isdigit( ) False >>>s2.isdigit( ) False >>>s3.isdigit( ) True >>>s4.isdigit( ) False >>>s5.isdigit( ) False

7	islower( )	Returns **True** if all the characters in the string arelowercase. **False** otherwise.	>>> s1.islower()
			True
			>>> s2.islower()
			True
			>>> s3.islower()
			False
			>>> s4.islower()
			False
			>>> s5.islower()
			True

8	isupper( )	Returns **True** if all the characters in the string areuppercase. **False** otherwise.	>>> s1.isupper()
			False
			>>> s2.isupper()
			False
			>>> s3.isupper()
			False
			>>> s4.isupper()
			False
			>>> s5.isupper()
			False

9	isspace( )	Returns **True** if there are only	>>> " ".isspace()
			True
			>>> "".isspace()
			False

10	lower( )	Converts a string in lowercase characters.	>>> "HeLlo".lower() 'hello'
11	upper( )	Converts a string in uppercase characters.	>>> "hello".upper() 'HELLO'

12	lstrip( )	Returns a string after removing the leading characters. (Left side).
		if used without any argument, it removes theleading whitespaces.
			>>> str="data structure"
			>>> str.lstrip('dat')
			' structure'
			>>> str.lstrip('data')'
			structure'
			>>> str.lstrip('at')
			'data structure'
			>>> str.lstrip('adt')
			' structure'
			>>> str.lstrip('tad')
			' structure'

13	rstrip( )	Returns a string after removing the trailingcharacters. (Right side).
		if used without any argument, it removes thetrailing whitespaces.
			>>> str.rstrip('eur')
			'data struct'
			>>> str.rstrip('rut')
			'data structure'
			>>> str.rstrip('tucers')'data
			'

14	split( )	Splits the string from the specified separator and returns a list object with string elements.	>>> str="Data Structure"
			>>> str.split( )
			['Data', 'Structure']

Exercise Questions: String

1 Mark Questions		
Q.No	Question	Answer
1.	print the string “India” 10 times.	>>>”india”*10
2.	What is the output of the following code >>>‘a’ in “computer”	False
3.	What is the output of the following code Strg=”computer” print(Strg[0: 8 : 2]	‘cmue’
4.	What is the output of the following? print('INDIA'.capitalize())	“India”
5.	Which of the following is not valid string in Python?	(iii) “Hello’

	(i) “Hello” (ii) ‘Hello’ (iii) “Hello’ (iv) None of the above	
6.	Suppose word = ‘amazing’, the what will be word[: : -2]?	‘giaa’

2 Mark Questions		
Sr.	**Question**	**Answer**
1.	If you give the following for str1=”Hello” why does python report error str1[2]=’p’	String is immutable data type.So it does not support item assignment
2.	Identify the output of the following Python statements. x="Vidyalaya" y="Vidya" if(y in x): print(y)	Vidya

3.	Look at the code sequence and select the correct output
	str="KVS RO Jaipur"
	for i in str:
	if(i.isupper()==True):
	print(i.lower(),end="")
	if(i.islower()==True):
	print(i.upper(), end="")
		kvsrojAIPUR

4.	Find the correct output of the following >>>str="The planet earth looks like a blue marble from outer space" >>>print(str.find('marble',50))	-1

5.	Find the value stored in ctr at the end of this code snippet:
	mystr="Darjeeling Tea has a strong flavour"
	ctr=0
	for i in mystr:
	if i in 'aeiouAEIOU':
	ctr += 1
	print(ctr)
		12

3 Mark Questions		
Sr.	**Question**	**Answer**
1.	Write a Python program to input a line of text and a character from user to print the frequency of the character in the line. For example Line entered is : this is a golden pen The character to search : e Then output is : Frequency is 2	line=input ("Enter a line of text :") ch=input ("Enter a character to search ") k=line.count(ch) print ("Frequency is :",k)

2.	Write a Python Program to input a string to check whether it is a Palindrome string or not. (A Palindrome string is that which is same from both ends like – NITIN, MALAYALAM, PULLUP)	s=input("Enter a word :")
		print ("You entered :", s)
		length=len(s)
		rev=""
		for i in range (-1,-length-1,-1):
		rev=rev+s[i]
		if s==rev:
		print ("Yes, palindrome")

		else: print ("Not a palindrome")
3.	Using string replication techniques print the following pattern using any loop. Hello Hello Hello Hello Hello Hello	for a in range(1,4): print("hello " * a)

4 Mark Questions		
Sr.	**Question**	**Answer**
1.	Find Output: my_string = 'Jhunjhunu' print(my_string[:3]) for i in range(len(my_string)): print(my_string[i].upper(),end="@") print() print (my_string) print (my_string[3:6])	Jhu J@H@U@N@J@H@U@N@U@ Jhunjhunu Njh
2.	Consider the following string mySubject: mySubject = "Computer Science" What will be the output of the following string operations : print(mySubject[0:len(mySubject)]) print(mySubject[-7:-1]) print(mySubject[::2]) print(mySubject[len(mySubject)-1]) print(2*mySubject) print(mySubject[::-2])	Computer Science Scienc Cmue cec e Computer ScienceComputer Science eniSrtpo
3.	Consider the following string country: country= "Great India" What will be the output of the following string operations(Any Four):- a) print(country[0:len(country)]) b) print(country[-7:-1]) c) print(country[::2]) d) print(country[len(country)-1]) e) print(2*country) f) print(country[:3] + country[3:])	a)Great India b)t Indi c)GetIda d)a e)Great IndiaGreat India f)Great India

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

Lists in Python

List is a standard data type of Python. It is a sequence which can store values of any kind. In Python, list and dictionary are mutable data types.

- List is represented by square brackets " [ ] "
  For ex -
- [ ] Empty list
- [1, 2, 3] integers list
- [1, 2.5, 5.6, 9] numbers list (integer and float)
- ['a', 'b', 'c'] characters list
- ['a', 1, 'b', 3.5, 'zero'] mixed values list
- ['one', 'two', 'three'] string list

Access Items From A List

List items can be accessed using its index position.
List is a sequence like a string.

•List also has index of each of its element.

•Like string, list also has 2 index, one for forward indexing (from 0, 1, 2, 3, ...to n-1) and one for backward indexing(from -n to -1).

•In a list, values can be accessed like string.
Example:

L=[10,20,30,40,50,60,70,80,90,100]

0	1	2	3	4	5	6	7	8	9	← Forward Indexing
10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**	
-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	← Backward Indexing

>>>L[2] Shows : 30
>>>L[-1] Shows : Last element of list i.e. 100.

Traversal of a list : Traversal of a list means to access and process each and every element of that list. Traversal of a list is very simple with for loop –

```
L=[10,20,30,40,50,60,70,80,90,100]
for i in L :
    print(i)
```

10
20
30
40
50
60
.
.

Operators in List :

A).**Concatenation (+ Operator):** Python allows us to join two or more lists using

Concatenation operator depicted by the symbol +.
Example :

```
>>> list1 = [1,3,5,7,9]
>>> list2 = [2,4,6,8,10]
>>>list1+list2    # + operator
[1,3,5,7,9,2,4,6,8,10]
```

B). **Repetition/ Replication Operator (* operator):** Python allows us to replicate a list using repetition operator depicted by symbol *.
Example:

```
>>> s = ['Hello']
>>> s * 4
['Hello', 'Hello', 'Hello', 'Hello']
```

C). **Membership(in / not in operator) :** Like strings, the membership operators in checks if the element is present in the list and returns True, else returns False.
Example:

```
>>> list1 = ['Red', 'Green', 'Blue']
>>> 'Green' in list1
True
>>> 'Cyan' in list1
False
```

The not in operator returns True if the element is not present in the list, else it returns False. Example:

```
>>> list1 = ['Red', 'Green', 'Blue']
>>> 'Cyan' not in list1
True
>>> 'Green' not in list1
False
```

Slicing : Like strings, the slicing operation can also be applied to lists. List elements can be accessed in subparts.
>>>list_name[start:stop:step]
Examples:

```
l=[10,20,30,40,50,60,70,80,90,100]
>>> l[2:6:1]
[30, 40, 50, 60]
>>> l[0:20:2]
[10, 30, 50, 70, 90]
>>> l[-8:-1:1]
[30, 40, 50, 60, 70, 80, 90]
>>> l[ : 6:1]
[10, 20, 30, 40, 50, 60]
>>> l[2 :  : 2]
[30, 50, 70, 90]
>>> l[ : : -1]
[100, 90, 80, 70, 60, 50, 40, 30, 20, 10]
>>> l[1:6:-1]
[]
>>> l[ : : 2]
[10, 30, 50, 70, 90]
```

List Methods and Built-in Functions :- Python provides some built-in functions for list manipulation .

>>>List_name.functionname()

Ex : **l=[10,20,30,40,50,60,70,80,90,100]**

Function/Method Name	Description	Example
len(list)	Returns number of elements in given list.	>>>len(l) 10
list(sequence)	It converts a sequence into list format.	>>>list("python") ['p','y','t','h','o','n']

List.index(<item>)	Returns the index of passed items.	>>> list1 = [10,20,30,20,40,10]
		>>> list1.index(20)
		1
		>> list1.index(90)
		ValueError: 90 is not in list

List.append(<item>)	Appends a single element passed as an
	argument at the end of the list
	The single element can also be a list
		>>> list1 = [10,20,30,40]
		>>> list1.append(50)
		>>> list1
		[10, 20, 30, 40, 50]
		>>> list1 = [10,20,30,40]
		>>> list1.append([50,60])
		>>> list1
		[10, 20, 30, 40, [50, 60]

List.extend(<list>)	Append the list (passed in the form of argument) at the end of list with which function is called.	>>> list1 = [10,20,30]
		>>> list2 = [40,50]
		>>> list1.extend(list2)
		>>> list1
		[10, 20, 30, 40, 50]

List.insert(<pos>, <item>)	Insert the passed element at the passed position.	>>> list1 = [10,20,30,40,50]
		>>> list1.insert(2,25)
		>>> list1
		[10, 20, 25, 30, 40, 50]
		>>> list1.insert(0,5)
		>>> list1
		[5, 10, 20, 25, 30, 40, 50]

List.pop(<index>)	Delete and return the element of passed index. Index passing is optional, if not passed, element from last will be deleted.	>>> list1 = [10,20,30,40,50,60]
		>>> list1.pop(3)
		40
		>>> list1
		[10, 20, 30, 50, 60]
		>>> list1 = [10,20,30,40,50,60]
		>>> list1.pop()
		60

List.remove(<value>)	It will delete the first occurrence of passed value but does not return the deleted value.	>>> list1 = [10,20,30,40,50,30]
		>>> list1.remove(30)
		>>> list1
		[10, 20, 40, 50, 30]
		>>> list1.remove(90)
		ValueError:list.remove(x):x not in list

List.clear ( )	It will delete all values of list and gives an empty list.	>>> list1 = [10,20,30,40,50,30]
		>>> list1.clear()
		>>> list1
		[]

List.count (<item>)	It will count and return number of occurrences of the passed element.	>>> list1 = [10,20,30,10,40,10]
		>>> list1.count(10)
		3
		>>> list1.count(90)
		0

List.reverse ( )	It will reverse the list and it does not create a new list.	>>> list1 = [34,66,12,89,28,99]
		>>> list1.reverse()
		>>> list1
		[99, 28, 89, 12, 66, 34]

List.sort ( )	It will sort the list in ascending order. To sort the list in descending order, we need to write----- list.sort(reverse =True).	>>>list1 =
		['Tiger','Zebra','Lion',
		'Cat', 'Elephant' ,'Dog']
		>>> list1.sort()
		>>> list1
		['Cat', 'Dog', 'Elephant', 'Lion',
		'Tiger', 'Zebra']
		>>> list1 = [34,66,12,89,28,99]
		>>> list1.sort(reverse = True)
		>>> list1
		[99,89,66,34,28,12]

List.sorted()	It takes a list as parameter and creates a
	new list consisting of the same elements
	arranged in sorted order
		>>> list1 = [23,45,11,67,85,56]
		>>> list2 = sorted(list1)
		>>> list1
		[23, 45, 11, 67, 85, 56]
		>>> list2
		[11, 23, 45, 56, 67, 85]

min()
 	
max()
 	
sum()
	Returns minimum or smallest element
	of the list
	
	Returns maximum or largest element of
	the list
	
	Returns sum of the elements of the list
		>>> list1 = [34,12,63,39,92,44]
		>>> min(list1)
		12
		>>> max(list1)
		92
		>>> sum(list1)
		284

Nested Lists

When a list appears as an element of another list, it is called a nested list.
Example :

```
>>> list1 = [1,2,'a','c',[6,7,8],4,9]
>>> list1[4]
[6, 7, 8]
```

To access the element of the nested list of list1, we have to specify two indices list1[i][j]. The first index I will take us to the desired nested list and second index j will take us to the desired element in that nested list.

Some Programs on List

 # find the max value in a list

```
l=[]
```

```
n=int(input("Enter number of elements:"))
for i in range(1,n+1):
        b=int(input("Enter element:"))
        l.append(b)
        l.sort()
        print("Largest element is:",l[n-1])
```

Run:
Enter number of elements: 4
Enter element: 10
Enter element: 5
Enter element: 8
Enter element: 9
Largest element is: 10

 # find the mean of a list

```
l=[]
n=int(input("Enter number of elements:"))
for i in range(1,n+1):
   b=int(input("Enter element:"))
   l.append(b)
avg=sum(l)/n
print("Average:",avg)
```

Run:
Enter number of elements: 4
Enter element: 10
Enter element: 5
Enter element: 7
Enter element: 2
Average: 6.0

*** Frequency of an element in list**

```
my_list= [101,101,101,101,201,201,201,201]
print("Original List : ",my_list )
n=int(input("enter the element which you want to count:"))
print( my_list.count(n))
```

Run:
Original List : [101, 101, 101, 101, 201, 201, 201, 201]
enter the element which you want to count: 201
4

Exercise Questions: List Manipulation

1 Mark Questions		
Sr.	Question	Answer
1.	Suppose a list is L=[2, 33, "KVS", 14, 25], what is L[-3]?	"KVS"
2.	Find output: List1=[13,18,16,16,13,18] print(List1.index(16))	2
3.	Given a list L=[1, 2, ["COMPUTER", "SCIENCE"], "IS", "TUPLE"] What will be the value of L[- 3][1]	"SCIENCE"

4.	What is the output when we execute list("hello")?	['h','e','l','l','o']

2 Mark Questions		
Sr.	**Question**	**Answer**

1.	Find the output of the following Python Code:
	>>> L1=[10,20,30]
	>>> L2=[110, 220, 330]
	>>> L3=L1+L2
	>>> L4=L3[0:4]
	>>> print (L4)
	>>> L4[0]=L4[0]*10
	>>> L4[2]=L4[1]*5
	>>> L4[1]=L4[2]
	>>> L4[3]=L4[3] - 10
	>>> print (L4)
		[10, 20, 30, 110]
		
		[100, 100, 100, 100]

2.	How the pop( ) function is different from remove( ) function working with list in python ? Explain with example.	pop() function removes the last value and returns the same. >>>l=[10,20,30,20] >>>l.pop() 20 The *remove()* method removes the first matching value from the list. >>>l.remove(20)

3.	Write a Python program to find and display those place names, in which there are more than 5 characters.
	For example :
	If the list l= ["Miao", "Tawang", "Chabua", "Kimin", "Imphal", "Dimapur","Goa"]
	The following should get displayed :
	Tawang
	Chabua
	Imphal
	Dimapur
		l=["Miao", "Tawang", "Chabua", "Kimin", "Imphal", "Dimapur","Goa"]
		for i in l:
		  if(len(i)>=5):
		  print(i)

4.	What is the output when following code is executed ? >>>names = ['Amir', 'Bear', 'Charlton', 'Daman'] >>>print(names[-1][-1])	N

3 Mark Questions		
Sr.	**Question**	**Answer**

1.	Write a program that will take a number from the key board and find its presence in the list [10,20,30, 40,50,60]. It will print "Availabe" or "Not available"	List1=[10,20,30,40,50,60]
		Num=int(input("enter a number"))
		if Num in List1:
		  print("Available")
		else:
		  print("Not Available")

2.	What is the output when following code is executed ? names1 = ['Amir', 'Bear', 'Charlton', 'Daman'] names2 = names1	12

	names3 = names1[:]
	names2[0] = 'Alice'
	names3[1] = 'Bob'
	sum = 0
	for ls in (names1, names2, names3):
	  if ls[0] == 'Alice':
	  sum += 1
	  if ls[1] == 'Bob':
	  sum += 10
	print (sum)	

3.	*Write a program to check if a number is present in the list or not. If the number is present, print the position of the number. Print an appropriate message if the number is not present in the list.*	lst = eval(input("Enter first list :-"))
		num = int(input("Enter the number which you want to search :-"))
		if num in lst :
		  print(lst.index(num))
		else :
		  print("Number not found")

4.	*Crate the following lists using a for loop:*
	
	(a). A list consisting of the integers 0 through 49.
	
	(b). A list consisting the square of the integer 1 through 50
		a)
		lst = [ ]
		for i in range(50):
		  lst = lst + [i]
		print(lst)
		b) lst = [ ]
		for i in range(51):
		  lst = lst + [i**2]
		print(lst)

5.	*Write a program to increment the elements of a list with a number.*	lst = [ ]
		while True :
		  num = int(input("Enter a number :"))
		  lst.append(num)
		  ch = input("for quit enter y or Y =")
		  if ch == "Y" or ch=='y':
		  print(lst)
		  break

4 Mark Questions		
Sr.	**Question**	**Answer**

1.	Write a Python program to input 10 numbers to store in the list and print the third largest number.
	For example, if the entered numbers in the list are List are
	36, 25, 14, - 951, 75, - 85, 654, 88, 9521, 657, then output will be
	The third largest number is : 654
		L=list()
		for i in range (10):
		  k=int(input("Enter a number :"))
		  L.append(k)
		L.sort()
		print ("List is ", L)
		print ("The third largest number is :", L[-3])

2.	Create the following lists using a for loop:
	(a) A list containing of the integers 0 through 49.
	
	(b) A list containing squares of the integers 1 through 50.
		(a)>>>L=list()
		>>> for i in range (50):
		  L.append(i)
		>>> print (L)
		(b) >>>L=list()
		>>> for i in range (51):
		  L.append(i*i)
		>>> print(L)

3.	Find the output of the following code:	

	>>> L=["These", "are", "a", ["few", "words"], "that", "we", "will", "use"]
	>>> print (L[3:4])
	>>> print (L[3:4][0][1])
	>>> print (L[3:4][0][1][2])
	>>> print ("few" in L)
	>>> print ("few" in L[3])
	>>> print (L[0::2])
	>>> print (L[4:])
	>>> print (L)
		[['few', 'words']]
		words
		r
		False
		True
		['These', 'a', 'that', 'will']
		['that', 'we', 'will', 'use']
		['These', 'are', 'a', ['few', 'words'], 'that', 'we', 'will', 'use']

4.	Find and write the output of the following Python code : x= [1, 2, [3, "KVS", 4], "KV"] print(x[0]) print(x[2]) print(x[-1]) print(x[0:1]) print(2 in x) print(x[0]==8) print(len(x)) x.extend([12,32,4]) print(len(x))	1 [3, 'KVS', 4] KV [1] True False 4 7
5.	The record of a student (Name, Roll No., Marks in five subjects and percentage of marks) is stored in the following list: sr = ['Raman','A-36',[56,98,99,72,69], 78.8] Write Python statements to retrieve the following information from the list sr. (a) Percentage of the student (b) Marks in the fifth subject (c) Maximum marks of the student (d) Roll no. of the student (e) Change the name of the student from 'Raman' to 'Raghav'	(a)>>>sr[3] or >>>sr[-1] (b)>>>sr[2][4] (c)>>>max(sr[2]) (d)>>>sr[1] (e)>>>sr[0]="Raghav"