**1. \n and \t are known as \_\_\_\_\_\_\_\_\_\_.**

a) Escape Sequence

b) Special Character

c) Keyword

d) None of the above

**2.What is the error returned by the following code:**

str = "Learning"

print(str[10])

a) Error in Index

b) Index out of range in string

c) Index Error

d) None of the above

**3. Name the function which is used to find the length of string ?**

a) length( )

b) len( )

c) strlen( )

d) slen( )

**4. Which operator is used with integers as well as with strings?**

a) /

b) //

c) \*\*

d) \*

**5. What will be the output of the following?**

str = “Welcome"

l=len(str)

print(l)

a) Error b) 7

c) 6 d) 8

**6. What will be the output of below Python code?**

str1="Information"

print(str1[2:8])

a) format b) formatio

c) orma d) ormat

**List:**

* A list is a sequence of similar or different data types.
* It is an ordered collection of values (elements) enclosed within square brackets [ ].
* List can be modified.
* The position of an element in the list is indexed, first element starting with zero, second indexed as one and so on.
* The elements of list should be specified within square brackets.
* **Example:**

Marks = [10, 23,12, 41, 75]

Fruits = [“Apple”, “Orange”, “Mango”, “Banana”]

MyList = [ ]

Accessing elements:Index value can be used to access an element in a list.

* Index value can be a positive or negative integer number.
* **Example:**

Marks = [10, 23, 41, 75]

* Positive value of index counts from the beginning of the list and negative value means counting backward from end of the list (i.e. in reverse order).
* To access an element in a list, index of the element needs to be mentioned along with the name of the list.
* To access all elements from a list, loops are used. The initial value of the loop must be zero to access the element from beginning.
* **Example:** >>> Marks = [10, 23, 41, 75]

a) >>> print (Marks[0])

out: 10

b) >>> print (Marks[-1])

out: 75

c) >>> i = 0

while i < 4:

print (Marks[i],end=’\t’)

i = i + 1

10 23 41 75

* **Example:**

d) >>> i = -1

while i >= -4:

print (Marks[i])

i = i + -1

**Output:**

75

41

23

10

* Lists are mutable, which means they can be changed by using assignment operator.
* **Syntax:**

List\_Variable [index of an element] = New Value

List\_Variable [index from : index to] = New Values

* index to is the upper limit of the range which is excluded.
* For example, if the range is [0:5], Python takes only 0 to 4 as element index. Thus, if you want to update the range of elements from 1 to 4, it should be specified as [1:5].
* To access values in lists, use the square brackets for slicing along with the index or indices to obtain value available at that index. For example −
* ***Lets try it out***
* >>> list1 = ['physics', 'chemistry', 45, 40.2]
* >>> list2 = [1, 2, 3, 4, 5, 6, 7 ]
* >>>print ("list1[0]: ", list1[0])
* >>> print ("list2[1:5]: ", list2[1:5])
* >>>list = [ 'abcd', 786 , 2.23, 'john', 70.2 ]
* >>>tinylist = [123, 'joy']
* >>>list # Prints complete list
* >>>list[0] # Prints first element of the list
* >>>list[1:3] # Prints elements starting from 2nd till 3rd
* >>>list[2:] # Prints elements starting from 3rd element
* >>>list[:3] # Prints elements starting from beginning till 3rd [0:1] [ :3]
* >>>list[1:-1] # Prints all elements except the first and last
* >>>tinylist \* 2 # Prints list two times
* >>>list + tinylist # Prints concatenated lists
* >>>len(list) # Prints length of the list

**Python Lists- Updating and Deleting from Lists**

* Unlike the other datatypes we played with so far, lists are mutable. Which means, we can actually change the value of lists .

>>>list = [ 'abcd', 786 , 2.23, 'john', 70.2 ]

>>> list[0] = 111 #Modify the value of the 0th element

>>> list[1:3]=[‘Sam’,2000,300.89] #Modifies the value of 1st to 3rd element of the list.

>>> list[1:3]=[] #Deletes from 1st to the 3rd element, but doesn’t take the 3rd element.

>>> list[:] =[] #Clears complete list

p = [1, q, 4]

len(p)

* Method 1 : Using list concatenation

>>> a=a+[10,20]

This will create a second list in memory which can (temporarily) consume a lot of memory when you’re dealing with large lists

* Method 2 : Using append

Append takes a single argument(any datatype) and adds to the end of list

>>>list1=[10,20,30]

>>>list1.append(‘new’)

>>> list1

[10,20,30,’new’]

>>> list1.append([1,2,3])

>>>list1

[10,20,30,’new’,[1,2,3]]

* Method 3 : Using extend

extend takes a single argument(list),and adds each of the items to the list

>>>a=[10,20,30]

>>>a.extend([1,2,3])

>>>a

[10,20,30,1,2,3]

* Method 4 :Using insert

Insert can be used to insert an item in the desired place

>>>a=[10,20,30]

>>>a.insert(0,‘new’)

>>> a

[’new’,10,20,30]

>>>a.insert(100,‘python’)

>>> a

[’new’,10,20,30,’python’]

* We have already discussed deletion of lists in one of the sections before
* Deletion was done by selecting the exact index which we wanted to delete.
* For e.g a[1,2,3] is a list, and to delete the second element of the list we simply used the statement,

a[1]=[]

* There is also a built in method called remove(), which can be used to remove elements from a list.
* Here’s how we can use this Superpower ☺
* list.remove(obj) : Removes an object from the list

>>> a = [-1, 1, 66.25, 333, 333, 1234.5]

>>> a.remove(200)

Traceback (most recent call last):

File "<pyshell#4>", line 1, in <module>

a.remove(200)

ValueError: list.remove(x): x not in list

>>> a.remove(333) # removes the first matching value

>>> print(a) [-1, 1, 66.25, 333, 1234.5]

* Count : Returns count of how many times an object exists in the list.
* a = [66.25, 333, 333, 1, 1234.5]

print(a.count(333), a.count(66.25), a.count(’x’))

* Index : Returns the index of the occurrence of the first given value in the list

>>> a=[1,2,2,3]

>>> a.index(2)

1

>>> a.index(500) #Returns ValueError if not in list.

Traceback (most recent call last):

File "<pyshell#19>", line 1, in <module>

a.index(500)

ValueError: 500 is not in list

* Reverse : Reverses objects of list in place

>>> a.reverse()

>>> a

[3, 2, 2, 1]

**Descending order :**

p=[1,12,4,3,5]

p.sort(reverse=True)

Sort:

a=[2,4,5,3]

a .sort()

a

l3.pop(2)#will remove item at 2nd position

l3.pop()#will remove last item

| Method | Description |
| --- | --- |
| list.append(obj) | Appends object obj to list |
| list.count(obj) | Returns count of how many times obj occurs in list |
| list.extend(seq) | Appends the contents of seq to list |
| list.index(obj) | Returns the lowest index in list that obj appears |
| list.pop(listindex) | Returns the removed element whose in index is given |
| list.remove(obj) | Removes object obj from list |
| list.reverse() | Reverses objects of list in place |
| list.sort() | Sorts objects inside a list |

| Method | Description |
| --- | --- |
| list.append(obj) | Appends object obj to list |
| list.count(obj) | Returns count of how many times obj occurs in list |
| list.extend(seq) | Appends the contents of seq to list |
| list.index(obj) | Returns the lowest index in list that obj appears |
| list.pop(listindex) | Returns the removed element whose in index is given |
| list.remove(obj) | Removes object obj from list |
| list.reverse() | Reverses objects of list in place |
| list.sort() | Sorts objects inside a list |

Stimulants

1. Suppose listExample is [‘h’,’e’,’l’,’l’,’o’], what is len(listExample)?  
   a) 5  
   b) 4  
   c) None  
   d) Error
2. Suppose list1 is [2445,133,12454,123], what is max(list1) ?  
   a) 2445  
   b) 133  
   c) 12454  
   d) 123
3. Suppose list1 is [3, 5, 25, 1, 3], what is min(list1) ?  
   a) 3  
   b) 5  
   c) 25  
   d) 1
4. Suppose list1 is [1, 5, 9], what is sum(list1) ?

a) 1  
b) 9  
c) 15  
d) Error

5. Suppose list1 is [2, 33, 222, 14, 25], What is list1[-1] ?  
a) Error  
b) None  
c) 25  
d) 2

6. Suppose list1 is [2, 33, 222, 14, 25], What is list1[:-1] ?  
a) [2, 33, 222, 14].  
b) Error  
c) 25  
d) [25, 14, 222, 33, 2].

7. What is the output when following code is executed ?

>>>names = ['Amir', 'Bear', 'Charlton', 'Daman']

>>>print(names[-1][-1])

a) A  
b) Daman  
c) Error  
d) n

8. What is the output when the following code is executed ?

names1 = ['Amir', 'Bear', 'Charlton', 'Daman']

names2 = names1

names3 = names1[:]

names2[0] = 'Alice'

names3[1] = 'Bob'

sum = 0

for ls in (names1, names2, names3):

print(ls)

if ls[0] == 'Alice':

sum += 1

if ls[1] == 'Bob':

sum += 10

print( sum)

a) 11  
b) 12  
c) 21  
d) 22

9. What is the output when the following code is executed ?

>>>list1 = [11, 2, 23]

>>>list2 = [11, 2, 2]

>>>list1 < list2 is

a)True  
b) False  
c) Error  
d) None

10. To add a new element to a list we use which command ?  
a) list1.add(5)  
b) list1.append(5)  
c) list1.addLast(5)  
d) list1.addEnd(5)

11. To insert 5 to the third position in list1, we use which command ?

a) list1.insert(3, 5)  
b) list1.insert(2, 5)  
c) list1.add(3, 5)  
d) list1.append(3, 5)

12. What is the output when the following code is executed ?

myList = [1, 5, 5, 5, 5, 1]

max = myList[0]

indexOfMax = 0

for i in range(1, len(myList)):

if myList[i] > max:

max = myList[i]

indexOfMax = i

>>>print(indexOfMax)

a) 1

b) 2

c) 3

d) 4

13. To which of the following the “in” operator can be used to check if an item is in it?  
a) Lists  
b) Dictionary  
c) Set  
d) All of the mentioned

14. What will be the output?

veggies = ['carrot', 'broccoli', 'potato', 'asparagus']

veggies.insert(veggies.index('broccoli'), 'celery')

print(veggies)

1. [‘carrot’, ‘celery’, ‘potato’, ‘asparagus’].
2. [‘carrot’, ‘celery’, ‘broccoli’, ‘potato’, ‘asparagus’]
3. [‘carrot’, ‘broccoli’, ‘celery’, ‘potato’, ‘asparagus’].
4. [‘celery’, ‘carrot’, ‘broccoli’, ‘potato’, ‘asparagus’].

Home work:

1. Write a python program that creates a list of numbers from 1 to 40 that are divisible by 4 and print the list.
2. Write a python program to take input for two lists from the user and print the element which is common in both the lists.

## List Comprehension

List comprehension offers a shorter syntax when you want to create a new list based on the values of an existing list.

Example:

Based on a list of fruits, you want a new list, containing only the fruits with the letter "a" in the name.

Without list comprehension you will have to write a for statement with a conditional test inside:

fruits = ["apple", "banana", "cherry", "kiwi", "mango"]

newlist = []

for x in fruits:

if "a" in x:

newlist.append(x)

print(newlist)