

Que. Understanding how to create and access elements in a list.

Ans.

List is a collection of similar and dis-similar data elements. it is ordered, indexable, mutable it means we can change its values. List is created by using [] square brackets.

How to create list : by using square brackets we can create list

ex. l1 = [ ]

access elements in list : for accessing element in list we can use for loop

for i in l1:

print(i)

Que. Indexing in lists (positive and negative indexing).

Ans.

list indexing starts from 0. And in negative indexing -1 means last element of list

ex.

l1 = [ 1 , 2 , 3 , 4 ]

positive	0	1	2	3
----------	---	---	---	---

negative	-4	-3	-2	-1
----------	----	----	----	----

Que. Slicing a list: accessing a range of elements.

Ans.

using list slicing we can extract a specific portion from list. For list slicing we have to enter starting index value and ending index value and steps.

starting index value include in result and ending index value not included in result and steps means it specifies the interval between elements.

syntax : list[start:end:step]

Que. Common list operations: concatenation, repetition, membership.

Ans.

Concatenation :

it means joint two list we can concat two list using “+” operator.

```
L1 = [1,2,3,4]
```

```
L2 = [5,6,7,8]
```

```
L3 = L1 + L2
```

```
Print(L3)
```

```
Output : [1,2,3,4,5,6,7,8]
```

Repetition:

For repetition of list we can use for loop.

```
l1 = [1,2,3,4,5]
```

```
for i in range(1,4):
```

```
    print(l1)
```

membership :

it means data element is member of list or not. Using “in” membership operator

```
L1 = [1,2,3,4,5]
```

```
If 2 in L1:
```

```
    Print(True)
```

Que. Understanding list methods like append(), insert(), remove(), pop().

Ans.

append():

using append method we can add one data element at last of list.

Syntax: list.append()

Insert():

Using insert method we can add element to list at specific position by define index.

Syntax: list.insert(index,element)

Remove():

Using remove method we can remove data element from list by using element value.by default it will remove from last.

Syntax: list.remove(element)

Pop():

Using pop method we can remove data element from list by using index value.by default it will remove from last.

Syntax: list.pop(index)

Que. Basic list manipulations: addition, deletion, updating, and slicing.

Ans.

List is a collection of similar and dis-similar data elements. It is orderable , indexable and mutable. It means we can change its value.

Addition:

With the use of + operator. It will joint two list.

Deletion:

For deleting element from list we use remove() , pop() method and delete all element of list we use clear() method and delete whole list we use del method it will delete list from memory.

Updating:

For updating list, using of append() method we can add one element at end of list. Using of extend() method we can add more than one element at end of list. and with use of insert method we can add element at specific position in list.

Slicing:

In list slicing, we can extract a specific portion of list with use of starting and ending index value.

Starting index value consider in result and ending index value not consider in result value.

Que. Introduction to tuples, immutability.

Ans.

Tuple is a collection of similar and dis-similar data elements at single location. It is immutable it means we can change its value.

It is use when we want to ensure the integrity of data. Tuple is represent by ().

Que. Creating and accessing elements in a tuple.

Ans.

For creating tuple, tuple is create using parentheses “()” . for accessing element of tuple we use for loop like we used in list. It will iterate element of tuple.

Que. Basic operations with tuples: concatenation, repetition, membership.

Ans.

Concatenation :

it means joint two tuple we can concat two list using “+” operator.

```
t1 = (1,2,3,4)
```

```
t2 = (5,6,7,8)
```

```
t3 = t1 + t2
```

```
Print(t3)
```

```
Output : (1,2,3,4,5,6,7,8)
```

Repetition:

For repetition of tuple we can use for loop.

```
t1 = (1,2,3,4,5)
```

```
for i in range(1,4):
```

```
    print(t1)
```

membership :

it means data element is member of tuple or not. Using “in” membership

operator

```
t1 = (1,2,3,4,5)
```

```
if 2 in t1:
```

```
    Print(True)
```

Que. Accessing tuple elements using positive and negative indexing.

Ans.

Tuple indexing start from 0 and negative indexing start from -1.

```
T1 = (1,2,3,4,5,6,7)
```

```
Print(t1[0])    # 1
```

```
Print(t1[3])    # 4
```

```
Print(t1[2:5])  # 3,4,5
```

```
Print(t1[-2])   # 6
```

Que. Slicing a tuple to access ranges of elements.

Ans.

With use of slicing we can extract a specific portion of tuple.

```
T1 = (1,2,3,4,5,6,7)
```

```
Print(t1[::-1]) # 7,6,5,4,3,2,1
```

```
Print(t1[1:5])  # 2,3,4,5
```

Que. Introduction to dictionaries: key-value pairs.

Ans.

dictionary is a most powerful concept in python and most powerful concept of collection data type. it represent by {} brackets. It contain key and value in pair.

It is indexable , mutable - we can change its value but it contain unique keys, orderable.

Que. Accessing, adding, updating, and deleting dictionary elements.

Ans.

Accessing:

```
d = {"a" : 1 , "b" : 2 , "c" : 3}
```

```
Print(d["a"]) # 1
```

```
Print(d.get("a")) # 1
```

Updating:

```
d = {"a" : 1 , "b" : 2 , "c" : 3}
```

```
d["c"] = 3 # add new key value
```

```
d["a"] = 111 # updating into existing dictionary.
```

Deleting:

```
d = {"a" : 1 , "b" : 2 , "c" : 3}
```

```
del d["c"] # remove an item.
```

Que. Dictionary methods like keys(), values(), and items().

Ans.

Keys():

With use of keys function we can iterate keys of dictionary using for loop.

Values():

With use of values function we can iterate values of dictionary using for loop.

Items():

With use of items function we can iterate keys and values of dictionary using for loop.

Que. Iterating over a dictionary using loops.

Ans.

For iterating item , key and values of dictionary we use some pre-defined function or methods.

Items():

With using it we can iterate keys and values from dictionary.

```
d = {"a" : 1 , "b" : 2 , "c" : 3}
```

```
For k,v in d.items():
```

```
    Print(f"key = {k} and value = {v} ")
```

Values():

With using it we can iterate values from dictionary.

```
d = {"a" : 1 , "b" : 2 , "c" : 3}
```

```
For v in d.values():
```

```
    Print(f"value = {v} ")
```

items():

With using it we can iterate keys from dictionary.

```
d = {"a" : 1 , "b" : 2 , "c" : 3}
```

```
For v in d.keys():
```

```
    Print(f"key = {k} ")
```

Que. Merging two lists into a dictionary using loops or zip().

Ans.

Using for loop:

```
L1 = ["name", "age", "city"]
```

```
L2 = ["vishwajeetsinh", 24, "Ahmedabad"]
```

```
d = {}
```

```
For i in range(len(l1)):
```

```
    d[l1[i]] = l2[i]
```

using zip():

```
L1 = ["name", "age", "city"]
```

```
L2 = ["vishwajeetsinh", 24, "Ahmedabad"]
```

```
d = dict(zip(l1,l2))
```

Que. Counting occurrences of characters in a string using dictionaries.

Ans.

```
s = "vishwajeetsinh"
```

```
d = {}
```

```
for i in s:
```

```
    d[i] = s.count(i)
```

```
print(d)
```



Que. Defining functions in Python.

Ans.

Function is a block of code it run only when its is called.

It is define by def keyword.

Two types of function:

- 1) Built-in function.
- 2) User define function

Four categories of function:

- 1) Function with parameter and with return type.
- 2) Function with parameter and without return type.
- 3) Function without parameter and with return type.
- 4) Function without parameter and without return type.

Que. Different types of functions: with/without parameters, with/without return values.

Ans.

- 1) Function with parameter and with return type.

It takes input and return value.

- 2) Function with parameter and without return type.

It takes input but does not return value. We have to print.

- 3) Function without parameter and with return type.

It does not take any input but return value.

- 4) Function without parameter and without return type.

It does not take any input also does not return values.

Que. Anonymous functions (lambda functions).

Ans.

It is function without name. it is used with lambda keyword.

It accept number of arguments but return on expression.

Lambda argument : expression.

Que. Introduction to Python modules and importing modules.

Ans.

Modules are files containing python code , function , class , variable.

Two types of module:

- 1) Built-in module. (standard module)
- 2) 3<sup>rd</sup> party module.

We can use module in our code using import keyword.

Syntax = import module\_name

Que. Standard library modules: math, random.

Ans.

Math:

With use of math module we can do mathematical operations.

Math.sqrt()

Math.ceil()

Math.floor()

Math.sin()

Math.cos()

Mat.pi()

Random:

With use of random we can create random number or choice.

Random.randint(1,100)

L1 = [1,2,3,4,5,6]

Random.choice(L1)

Que. Creating custom modules.

Ans.

For create a module we have to first write function to on file.

Ex.

Hello.py

```
def fun():
```

```
    print("hello python")
```

import module:

```
import hello
```

```
hello.fun()
```

```
# print = hello python
```