Q-1 How memory is managed in Python?

Ans. According to the Python memory management documentation,python has a private heap that stores our programme objects and data structures. Python memory management work and allows us to concentrate on our code.

Types of memory allocation

There are two types of memory allocation in python:

1.static memory:

The stack data structure provides **static memory allocation**,meaning the variables are in the stack memory. Statically assigned variables, as the name implies, are permanent; this means that they must be allocated in advance and persist for the duration of the programme. Another point to remember is that we cannot reuse the memory allocated in the stack memory .Therefore, memory reusability is not possible.

Ex:

```
X=20
Y=[]
Z=""
```

2.dynamic memory

The **dynamic memory allocation** uses heap data structures in its implementation, implying that variable are in the heap memory. As the name suggest, dynamically allocated variables are not permanent and can be changed while a

program is running. Additionally,allotted memory can be relinquished are reused. However, it takes longer to complete because dynamic memory allocation occurs during progrmme execution. Furthermore, after utilizing the allocated memory ,we must release it. Otherwisae, problems such as memory leaks might arise.

Ex:

```
X=[0]*5 # this memory for 5 integers is allocated
```

Q- 2 what is the purpose continue statement in python?

Ans . the continue statement cancels every statement running in the iteration of a loop and moves the control back to the top of the loop. In simple words, it continues to the next iteration when a certain condition is reached.

We can use the **continue statement** in the **while** and **for** loops.

The continue statement in a for loop.

In the example below, we will use the continue statement to continue to the next iteration when the letter variable is equal to 0.

EX:

```
For letter in 'python':
    If letter=='0':
        Continue
```

Print(“current letter:”, letter)

The continue statement in a while loop

In the below, we will use the continue statement to continue to the next iteration when x equals 2.

EX:

```
X=0
While  x < 5:
X+=1
If x == 2:
Continue
Print(x)
```

Q-3 what are negative indexes and why are they used?

Ans. The sequences in python are indexed and it consist of\f the positive as well as negative numbers.

The numbers that are positive uses ‘0’ that is uses as first index and ‘1’ as the second index and the process goes on like that. The index for the negative number starts from ‘-1’ that represents the last index in the sequence and ‘-2’ As the penultimate index and the sequence carries forward like the positive number.

The negative index is used to remove any new-line spaces from the string and allow the string to except the last character that is given as S[:-1]. The negative index is also used to show the index to represent the string in correct order.

Q-4 what is list? How will you reverse a list?

Ans. Lists are used to store multiple items in a single variable. Lists are one of 4 built in data types in python used to store collection of data, the other 3 are tupple,set anf dictonary all with different qualities and usage. Lists are created using square brackets:

Ex:

```
mylist=[“banana”,”apple”,”mango”,]

Print(mylist)
```

List reverse() method python list class comes with the default reverse that inverts the order of items in the given list. Reversed() method it is python’s built im method that reverse all types sequnces such a list ,string,tuple etc.. slice oprator to reverse a list using for loop to reverse using list comprehension.

Q-5Differntive between append() and extend()?

Ans.

Append()	**Extend()**
The element passed as an argument is appended to the end of the list.	Each element of the iterable of passed as an argument gets appended to the end of the list.
An iterable passed as an argument appends without any change as a single element to the end of the list.	An iterable passed as an argument will append will append each of its elements to the end of the list.
Length of the list increase by 1.	Length of the list increases by the number of elements in the iterable.
Has a constant time complexity of 0(1).	Has a time complexity of 0 (k) where k is the length of the iterable.

Q-6 How to compare Two lists in python?

Ans. We will understand the different ways to compare two lists in python. We often come across situations wherein we need to compare the values of the data items stored in any

structure say list,tuple,string etc. comparison is the method of checking the data items of another list.

Methods to compare two lists in python

We can use either of the following methods to perform our comparision:

1.The reduce() and map()fuction

2.The collection.counter()function

3.Python sort() function along with == oprator

4.Python set() fuction along with == oprator

5.The difference() function

1. Python reduce() and map() functions

We can use the python map() fuction along with functools.reduce() function to compare the data items of two lists. The map() methd accepts a function and an iterable such as list, tuple,string as arguments. It applies the passed function to each item of the iterable and then return a map object i.e. an iterator as the result.

The fuctools.reduce() method applies the passed function to every element of the input iterable in a recursive manner.

Initially, it would apply the fuction on the first and the last second element and returns the result. The same process will continue on each of the elements until the list has no elemnts left.

As a combination, the map () function would apply the input function to every element and the reduce() fuction will make sure that it applies the fuction in a consecutive manner.

Ex:

l1=[10,20,30,40,50}

l2=[10,20,30,50,40,70]

l3=[10,20,30,40,50]

if functools.reduce(

Q- 7 what is tuple ? Differnce between list and tuple ?

Ans.

List	**Tuple**
List are mutable	Tuples are immutable.
Iterations are time- consuming.	Iterations are comparatively Faster.
Inserting and deleting items is easier with a list.	Accessing the elements is best accomplished with a tuple data type.
List consume more memory	Tuple consumes less than the list

List have several buit-in methods.	A tuple does not have many buit in methods because of immutability
A unexpected change or error is more likely to occur in a list.	In a tuple, changes and errors don't usually occur because of immutability.
Ex: fruits=['apple','banana','orange','mango']	Ex: Tuple=('rajkot','broda','bhuj','jamnagar')

Q- 8 how will you create dictonary using tuples in python ?

Ans. The tuple elements are enclosed inside the parenthesis, while dictionary elements are present in the form of a key- value pair and are enclosed between curly brackets .

We will show you how to convert a python tuple into a dictionary :

1 Using dict() function

2 Using dictionary Comprehension and enumerate() fuction

3 Using zip() and dict() functions

1 using dict() function

In python , use the dict() function to convert a tuple to a dictionary A dictionary object can be created with the dict() function. The dictionary is returned by the dict() method, which takes a tuple of tuples as an argument. A key – value pair is contained in each tuple.

Here in the below code, to create a dictionary , we used the dict() method and gave the dictionary comprehension as an argument . Dictionary comprehension is a technique for converting one dictionary into another. Elements from the original dictionary can be conditionally included in the new dictionary throughout this conversion, and each element can be converted as nedded.

The output is a key-value paired dictionary the first element of a tuple becomes a dictionary value.

Ex:

Inputtuple = ((5,”tutorialpoint”), (6, “python”), (7,”codes”))

Print (“the input tuple:”, inputtuple)

Resultdictionary = dict(x,y) for x ,y in inputuple)

Print(“the result dictionary:” ,resultdictionary)

Output:

The input tuple: ((5,”tutorialpoint”) , (6,”python”), (7, “codes”))

The resultdictionary : { 5: ‘tutorialpoin’ , 6 : ‘python’, 7 : ‘codes’}

2.Using dictionary comprehension and enumerate() fuction

Note : to convert two tuples into a dictionary , the tuples must be the same length otherwise , we won’t be able to match all of the key-value pairs.

Algorithms (steps)

Create two variables to store the first and second input tuples having the same length.

Use the if conditional statement, to check whether the length of both the tuples is the equal length or not.

Convert both the tuple into a dictionary using enumerate() function in a dictionary comprehension, if the condition is true.

Printing the result dictionary from the given two tuples after conversion .

The enumerate() method adds a counter to an iterable end returns the enumerate object.

Syntax: enumerate(iterablem, start=0)

Parameters:

Iterable – it can be any sequence / object/ iterable supporting iteration.

Start – enumerate() begins counting from this value. If start is not specified the value 0 is used.

Ex:

Inputtuple_1 = (‘tutorialpoint’,’python’,’codes’)

Inputtuple_2 –(5,6,7)

Print(“the inputtuple_1(keys) =“, inputtuple_1)

Print(“the inputtuple_2(values)=”, inputtuple_2)

If len(inputtuple_1) == len(inputtuple_2):

Resultdictionary ={(inputtuple_1[i] : inputtuple_2[i] for i, _in enumerate(inputtuple_2)}

Print(“the resultdictionary: “,resultdictionary)

Output:

The inputtuple_1(keys) = (‘tutorialpoint’, ‘pythpn’,’codes’)

The inputtuple_2(values) = (5,6,7)

The result dictionary : { ‘tutorialpoint’ : 5 , ‘pyhton’ :6, ‘codes’: 7}

3 Using zip() and dict() functions

Algorithms (steps)

Create two variables to store the first and second input tuples having the same length

Use the if conditional statement to check whether the length of both the tuple is the equal length or not.

Convert both the tuples into a dictionary using zip() and dict() fuctions, if the condition is true.

Zip() method : the zip () method takes iterables combnies them into a tuple, and returns it.

Syntax : zip(*iterables)

Dict() function: the dict() fuction is used to create a dictionary.

Printing the result dictionary from the given two tuples after conversion.

Ex:

Input Tuple_1 = (‘tutorialpoint’, ‘python’, ‘codes’)

Input Tuple _2 = (5,6,7)

Print(“the input Tuple_1(keys) =”, input Tuple_1)

Print(“the input Tuple_2(values)=”,input Tuple_2)

If len (input Tuple_1) == len(input Tuple_2):

ResutDictionary = dict(zip(inputTuple_1, input Tuple_2))

Print(“the resut dictionary : “, resultdictionary)

Output:

The input Tuple_1(keys) = (‘tutorialpoint’, ‘python’, ‘codes’)

The input Tuple_2(values) =(5,6,7)

The result dictionary = {‘tutorialpoint : ‘ 5, ‘python:’ 6, ‘code: ‘ ,7}

We learned how to convert a tuple to a dictionary using three diffrenet fuctions in this article. We also observed that if there are two tuples, we can create a dictionary by using two different by using two different methods,zip() and enumerate() to take the first tuple element as keys and second tuple elements as values() .

Q-9 why do you use the zip() method in python?

Ans. The zip() fuction returns a zip object, which is an iterator of tuples where the first item in each passed iterator is paired together, and then the second item in each passed iterator are paried together etc.

If the passed iterators have different lengths , the itrator with the least items decides the length of the new iterator.

Syntax:

Zip(iterable1, iterable2, iterable3….)

Parameters values

Iterator1, iterator obejcts that will be joined togher

Iterator2,

Iterator3,…..

EX:

a= (“komal”,”reema”,”meeta”)

b=(“python”,”php”,”java”)

x= zip(a,b)

output:

((‘komal ‘ , ‘pyhton’), (‘reema’, ‘php’), (‘meeta’, ‘java’))