

Q1. Can you create a programme or function that employs both positive and negative indexing? Is there any repercussion if you do so?

Q2. What is the most effective way of starting with 1,000 elements in a Python list? Assume that all elements should be set to the same value.

Q3. How do you slice a list to get any other part while missing the rest? (For example, suppose you want to make a new list with the elements first, third, fifth, seventh, and so on.)

Q4. Explain the distinctions between indexing and slicing.

Q5. What happens if one of the slicing expression's indexes is out of range?

Q6. If you pass a list to a function, and if you want the function to be able to change the values of the list—so that the list is different after the function returns—what action should you avoid?

Q7. What is the concept of an unbalanced matrix?

Q8. Why is it necessary to use either list comprehension or a loop to create arbitrarily large matrices?

A1. Yes, it is possible to create a program or function that employs both positive and negative indexing. There is no repercussion for doing so as long as the indexing is done correctly and within the bounds of the list or array being indexed.

A2. The most effective way of starting with 1,000 elements in a Python list and setting all elements to the same value is by using list comprehension. The following code can be used:

```
python
my_list = [0] * 1000
```

This will create a list of 1,000 elements with all elements set to 0.

A3. To slice a list and get every other part while missing the rest, you can use the step parameter in Python slicing. For example, to get a new list with the elements first, third, fifth, seventh, and so on, you can use the following code:

```
python
my_list = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
new_list = my_list[::2]
```

The `::2` in the slicing specifies to take every other element starting from the first.

A4. Indexing refers to accessing a single element in a list or array using its position or index, while slicing refers to accessing a range of elements in a list or array using their positions or indices.

A5. If one of the slicing expression's indexes is out of range, Python will not raise an error, but it will return an empty list or an empty slice.

A6. If you want a function to be able to change the values of a list after it returns, you should avoid creating a new list object inside the function and instead modify the original list object. You should also avoid using the `=` operator to assign a new value to the list parameter as this will create a new list object.

A7. An unbalanced matrix is a matrix where the number of rows is not equal to the number of columns.

A8. It is necessary to use either list comprehension or a loop to create arbitrarily large matrices because a matrix is essentially a two-dimensional list or array, and its size needs to be specified at the time of creation. List comprehension and loops provide a way to create lists or arrays with arbitrary sizes.