1. How memory is managed in Python?

Memory Management in Python:

Python uses a private heap space to manage memory. The Python memory manager handles the allocation of heap space for Python objects. It uses a system of private heaps, where all Python objects and data structures are stored. Python's memory manager is responsible for allocating memory for new objects and reclaiming memory that is no longer in use (garbage collection).

**2. What is the purpose continue statement in python?**

Purpose of continue statement in Python:

The continue statement is used inside loops (for, while) to skip the rest of the code inside the loop for the current iteration and proceed to the next iteration. When encountered, the continue statement causes the program to jump to the next iteration of the loop without executing the remaining code below it within the loop block. It is useful when you want to skip certain iterations based on a condition without terminating the entire loop.

**3. What are negative indexes and why are they used?**

Negative indexes in Python are used to access elements from the end of a sequence, such as a list or a string. The index -1 refers to the last element, -2 refers to the second-to-last element, and so on. Negative indexing provides a convenient way to access elements from the end without explicitly calculating the index.

For example, consider a list my\_list = [10, 20, 30, 40, 50]:

my\_list[-1] will return 50 (the last element).

my\_list[-2] will return 40 (the second-to-last element).

and so forth.

Negative indexing simplifies code when you need to access elements relative to the end of a sequence.