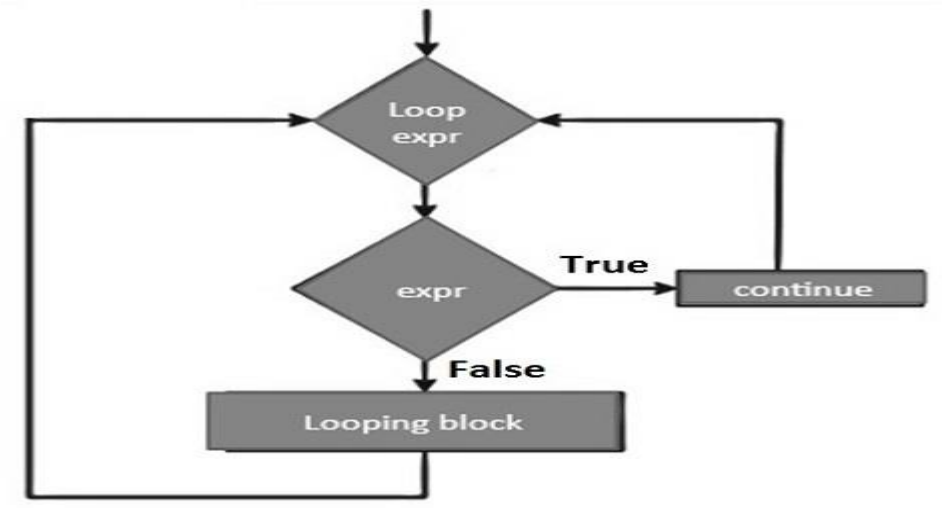
3.How memory is managed in Python?

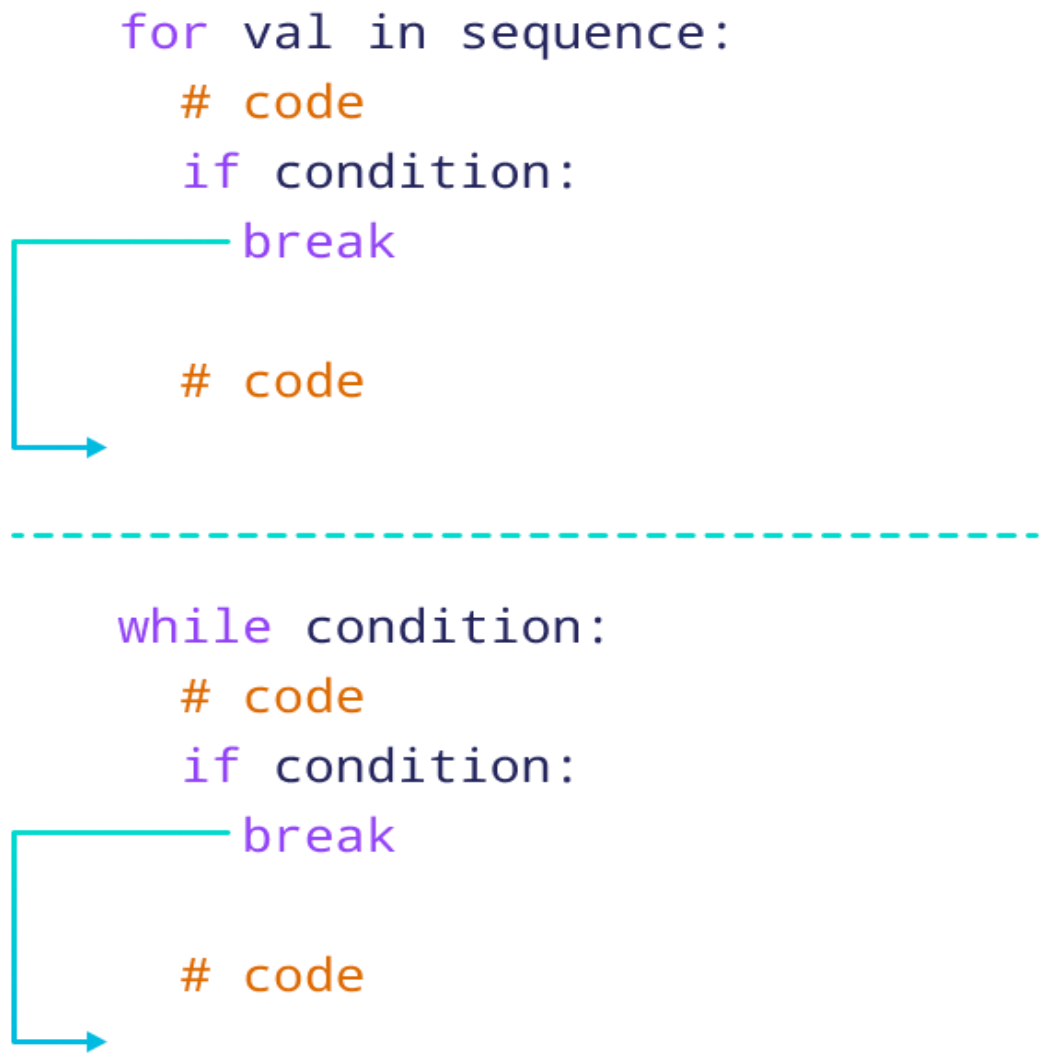
- In Python, memory is managed automatically. It uses reference counting to track how many references an object has. When the count drops to zero, the object's memory is freed. To handle circular references (objects referring to each other), Python has a garbage collector that finds and removes these unused objects. For efficiency, Python also uses memory pooling for small objects to reduce the overhead of frequent memory allocations. Developers generally don't need to worry about it, but tools are available to manage or optimize memory use if needed.

4.What is the purpose continue statement in python?

- Continue Statement in PythonThe continue statement is used to skip the remaining code inside a loop for the current iteration only. For instance, let's use continue instead of a break statement in the previous example.



- The continue statement in Python is used to skip the remaining code inside a loop for the current iteration and jump to the next iteration. It's typically used when you want to bypass certain parts of the loop body based on a condition, without exiting the loop completely.



- 14. What are negative indexes and why are they used?
- Negative indexes in Python allow you to access elements from the end of a sequence (like a list, tuple, or string). Instead of counting from the start (where the first element has an index of 0), negative indexing starts from the last element, which has an index of -1.