**🔹 1. Node Class**

class Node:

def \_\_init\_\_(self, val=None):

self.val = val

self.next = None

**🔍 What's Happening:**

* This defines a Node, which is the basic building block of a linked list.
* Each node has:
  + val: stores the data (e.g., 10, 20, etc.)
  + next: a pointer/reference to the **next node** in the list.
* Initially, next is None, meaning it doesn't point to anything yet.

**🔹 2. SinglyLinkedList Class**

class SinglyLinkedList:

def \_\_init\_\_(self):

self.head = None

**🔍 What's Happening:**

* This is the **linked list** itself.
* self.head keeps track of the **first node** in the list.
* Initially, the list is empty, so head is None.

**🔹 3. append() Method**

def append(self, data):

new\_node = Node(data)

if not self.head:

self.head = new\_node

else:

current = self.head

while current.next is not None:

current = current.next

current.next = new\_node

**🔍 What's Happening:**

**✅ Line-by-line:**

1. **new\_node = Node(data)**
   * Create a new node containing the data.
2. **if not self.head:**
   * Check if the list is empty (self.head is None).
3. **self.head = new\_node**
   * If the list is empty, make the new node the head.
4. **Else Block:**
   * The list already has at least one node.
   * Traverse to the end of the list:
   * current = self.head
   * while current.next is not None:
   * current = current.next
   * After the loop, current points to the **last node**.
   * Then:
   * current.next = new\_node
     + Link the last node to the new node.

**🔹 4. display() Method**

def display(self):

current = self.head

while current:

print(current.val, end=" -> ")

current = current.next

print("None")

**🔍 What's Happening:**

* Start from the head and follow the chain of .next pointers.
* Print each node’s value followed by ->.
* Stop when current becomes None (end of list).
* Finally, print "None" to mark the end.

**🔹 5. Putting It All Together (Sample Run)**

sll = SinglyLinkedList()

sll.append(10)

sll.append(20)

sll.append(30)

sll.display()

**🔍 Execution Trace:**

* sll = SinglyLinkedList()  
  → Creates an empty list.
* sll.append(10)  
  → Creates a node with 10 and sets it as head.
* sll.append(20)  
  → Traverses to end and adds node 20.
* sll.append(30)  
  → Adds node 30 to the end.
* sll.display()  
  → Prints:
* 10 -> 20 -> 30 -> None