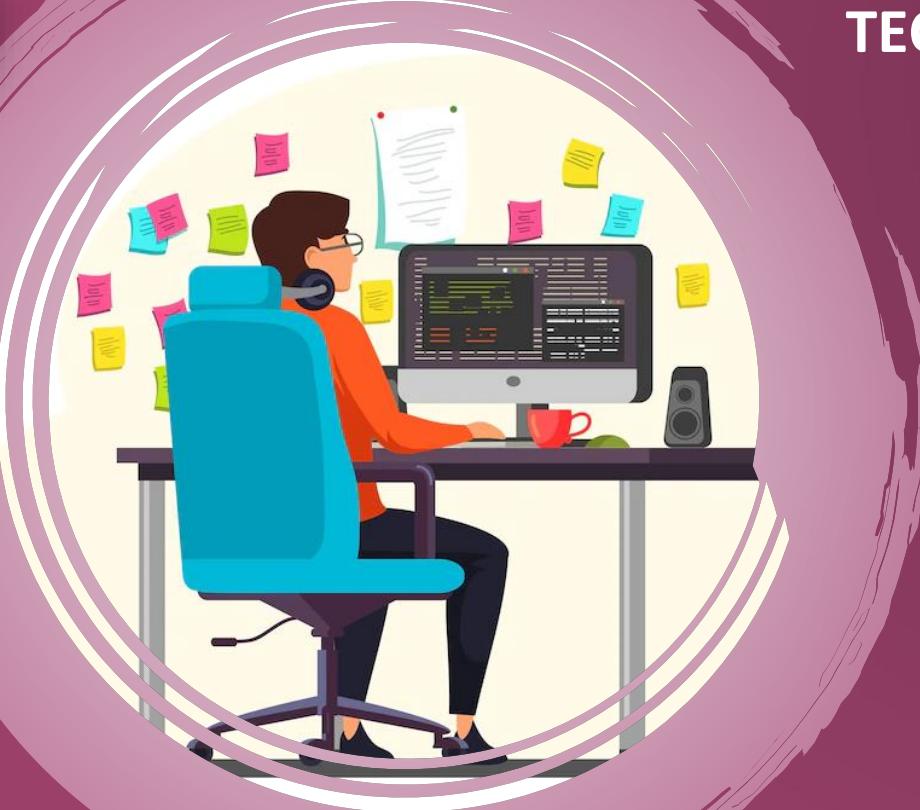




TAMILNADU ADVANCED
TECHNICAL TRAINING INSTITUTE



PYTHON

Linked List

A linked list is a sequence of data elements, which are connected together via links. Each data element contains a connection to another data element in form of a pointer. Python does not have linked lists in its standard library.

Types

- Simple Linked List
- Doubly Linked List
- Circular Linked List
- Doubly Circular Linked List
- Header Linked List

Basic operations

- Deletion
- Insertion
- Search
- Display

Application

- Linked Lists are used to implement stacks and queues.
- It is used for the various representations of trees and graphs.
- It is used in dynamic memory allocation(linked list of free blocks).
- It is used for representing sparse matrices.

Programs

```
class Node:  
  
    def __init__(self, data):  
  
        self.data = data # Assign data  
  
        self.next = None  
  
class LinkedList:  
  
    def __init__(self):  
  
        self.head = None
```

Programs

```
def printList(self):  
  
    temp = self.head  
  
    while (temp):  
  
        print(temp.data)  
  
        temp = temp.next  
  
    if __name__ == '__main__':  
  
        llist = LinkedList()  
  
        llist.head = Node(1)  
  
        second = Node(2)  
  
        llist.head.next = second
```

Programs

```
def push(self, new_data):  
  
    new_node = Node(data=new_data)  
  
    new_node.next = self.head  
  
    new_node.prev = None  
  
    if self.head is not None:  
  
        self.head.prev = new_node  
  
    self.head = new_node
```

Programs

```
def addBegin(last, data):  
  
    if (last == None):  
  
        return addToEmpty(last, data)  
  
    temp = Node(data)  
  
    temp.next = last.next  
  
    last.next = temp  
  
    return last
```

Programs

```
def addBegin(self, data):  
  
    if (self.last == None):  
  
        return self.addToEmpty(data)  
  
    temp = Node(data)  
  
    temp.next = self.last.next  
  
    self.last.next = temp  
  
    return self.last
```

Programs

```
def insertBegin(value):  
    global start  
  
    last = (start).prev  
  
    new_node = Node(0)  
  
    new_node.data = value  
  
    new_node.next = start  
  
    new_node.prev = last  
  
    last.next = (start).prev = new_node
```