# 📅 Day 29 — Introduction to Singly Linked List (SLL)

## 🎯 Learning Objectives

By the end of this lesson, you will:  
1. Understand what a singly linked list is.  
2. Learn the structure of nodes and pointers.  
3. Implement a basic singly linked list in Python.  
4. Compare it with arrays/lists.

Prerequisites: OOP’s

## 1. What is a Singly Linked List?

Definition: A singly linked list is a linear data structure where each element (node) contains data and a pointer to the next node.

## 2. Structure

Node:  
- data: stores the value  
- next: reference to the next node  
Visualization:  
[10 | next] → [20 | next] → [30 | next] → None

## 3. Why Linked List?

Advantages:  
- Dynamic size  
- Efficient insertion/deletion at head/tail

Disadvantages:  
- Slower access (no indexing like arrays)

## 4. Implementation in Python

 **(insert\_at\_beginning) Operation**

## 5. Advantages vs Arrays

|  |  |  |
| --- | --- | --- |
| Feature | Array/List | Singly Linked List |
| Memory | Fixed/Dynamic | Dynamic |
| Access Time | O(1) | O(n) |
| Insertion (middle) | O(n) | O(n) |
| Insertion (head) | O(n) | O(1) |
| Deletion | O(n) | O(1) at head |