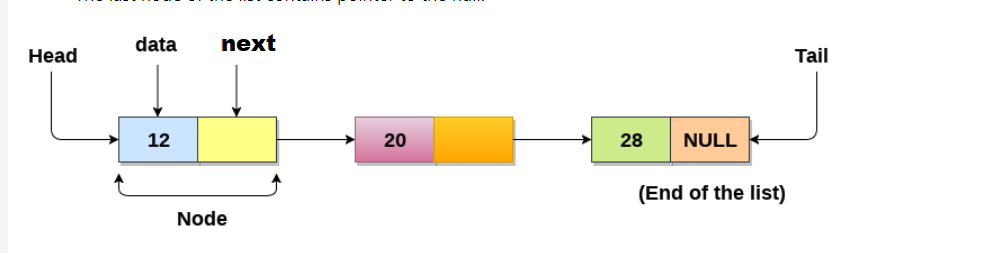
|  |
| --- |
| What is a Linked List?  A linked list is a linear data structure that consists of a series of nodes connected by pointers. Each node contains data and a reference to the next node in the list. Unlike arrays, linked lists allow for efficient insertion or removal of elements from any position in the list, as the nodes are not stored contiguously in memory. |
| [Linked Lists vs Arrays](https://www.geeksforgeeks.org/linked-list-vs-array/)  Linked List:   * Data Structure: Non-contiguous * Memory Allocation: Dynamic * Insertion/Deletion: Efficient * Access: Sequential   Array:   * Data Structure: Contiguous * Memory Allocation: Static * Insertion/Deletion: Inefficient * Access: Random |

|  |
| --- |
| [Types of Linked List](https://www.geeksforgeeks.org/types-of-linked-list/)   1. [Singly Linked List](https://www.geeksforgeeks.org/data-structures/linked-list/singly-linked-list/) 2. [Doubly Linked List](https://www.geeksforgeeks.org/doubly-linked-list/) 3. [Circular Linked List](https://www.geeksforgeeks.org/circular-linked-list/) 4. [Circular Doubly Linked List](https://www.geeksforgeeks.org/introduction-to-circular-doubly-linked-list/) 5. [Header Linked List](https://www.geeksforgeeks.org/header-linked-list-in-c/) |

|  |
| --- |
| What is Singly Linked List?  A singly linked list is a linear data structure in which the elements are not stored in contiguous memory locations and each element is connected only to its next element using a pointer. |
| Node, next element connected to Pointer |



Uses of Linked List

* The list is not required to be contiguously present in the memory. The node can reside any where in the memory and linked together to make a list. This achieves optimized utilization of space.
* list size is limited to the memory size and doesn't need to be declared in advance.
* Empty node can not be present in the linked list.
* We can store values of primitive types or objects in the singly linked list.