# LINKED LIST

Linked List is a linear data structure where each element is a separate object.

“+”:

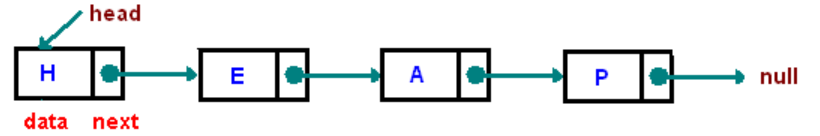
“-“:

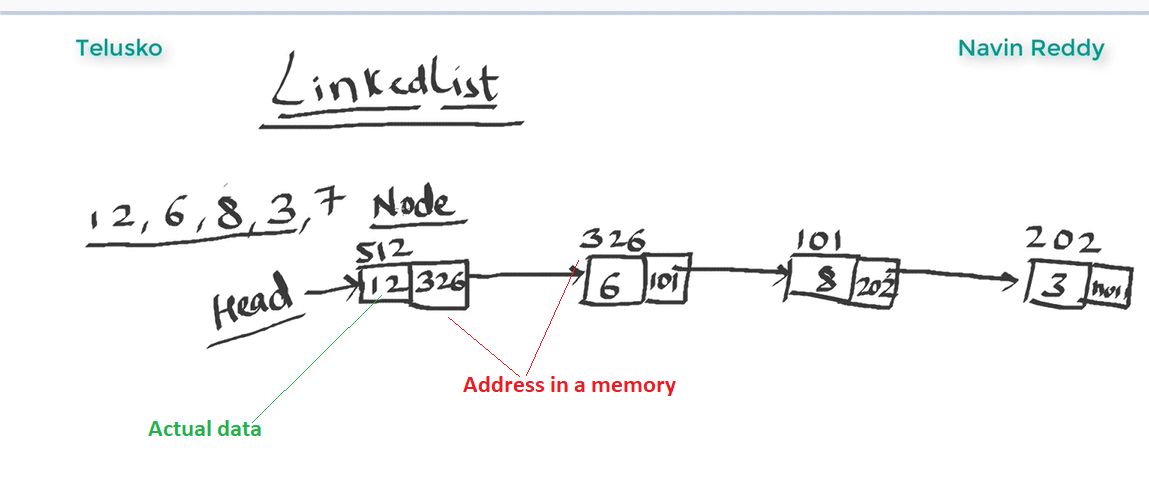
* One disadvantage of a linked list against an array is that it does not allow direct access to the individual elements. If you want to access a particular item then you have to start at the head and follow the references until you get to that item.
* Another disadvantage is that a linked list uses more memory compare with an array - we extra 4 bytes (on 32-bit CPU) to store a reference to the next node.

**Node** - Each element (we will call it a node) of a list is comprising of two items - the data and a reference to the next node. The last node has a reference to null.

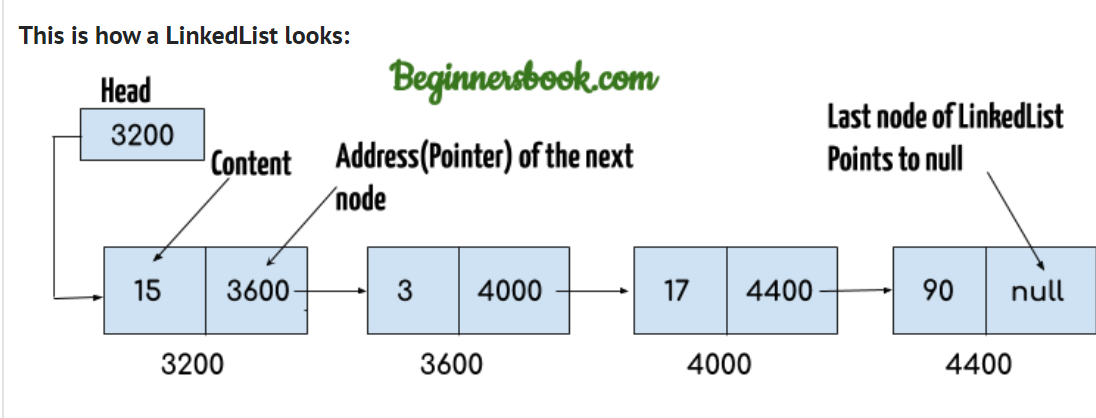
**Head** - The entry point into a linked list is called the head of the list. It should be noted that head is not a separate node, but the reference to the first node. If the list is empty then the head is a null reference.

The number of nodes in a list is not fixed and can grow and shrink on demand. The number of nodes in a list is not fixed and can grow and shrink on demand.





**Head** of the LinkedList only contains the Address of the **First element** of the List.



Arrays disadvantages:

Arrays are expensive to add new elements

LINKS:

Linked list

<https://www.cs.cmu.edu/~adamchik/15-121/lectures/Linked%20Lists/linked%20lists.html>