

# FND22

Both a stack and a linked list are considered ADTs. A stack implements a LIFO (last in, first out) approach to process data. In a stack when we want to add (push) we always add to the top. When we want to remove an item from a stack, we pop it. What gets popped is the last element that was pushed. All of this is to say that, in a stack, a new element is always stored at the top. Therefore, the functionalities of the stack are push and pop. Because of such functionalities, we can only manipulate what is present at the top of the stack. In a linked list, a new item is stored in the first free and available memory location. For a linked list, we add a new item (node) by insertion. We can remove a node by deleting it. This can be done by giving the key or the position of the node that we want to delete. We can search a linked list for an element. So, one of the biggest differences between a stack and a linked list is that in the stack we can only manipulate and read the element at the top. While, in a linked list we can manipulate and read any element regardless of its position in the list.