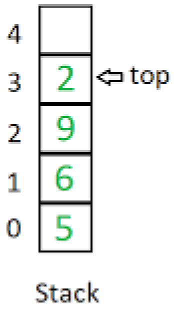
# Stack

**Stack**: Stack is a linear data structure in which insertion and deletion can take place only at one end called top. Stack works on the principle of Last in First out (LIFO).

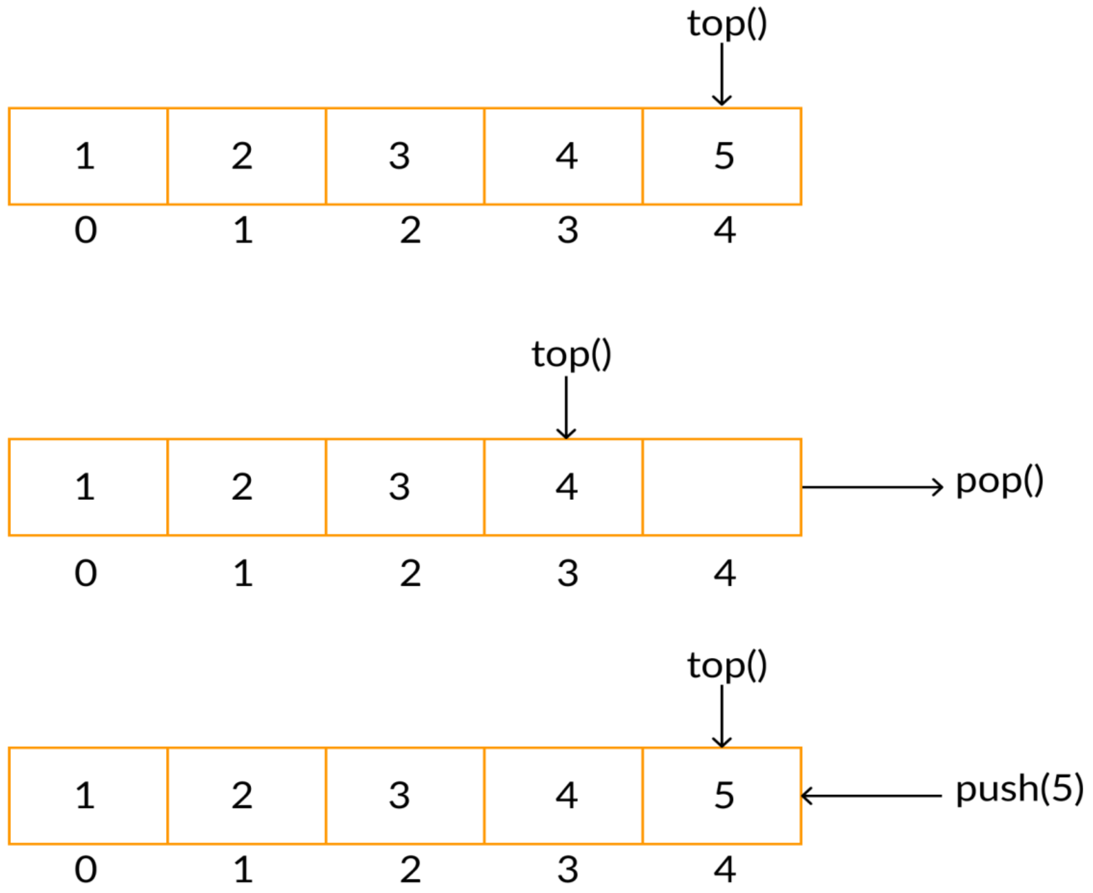
**Example**: Deck of cards, Piles of Plates.



### Representation of stacks

Stacks can be represented using Array and Linked List.

1. **Stacks using arrays**

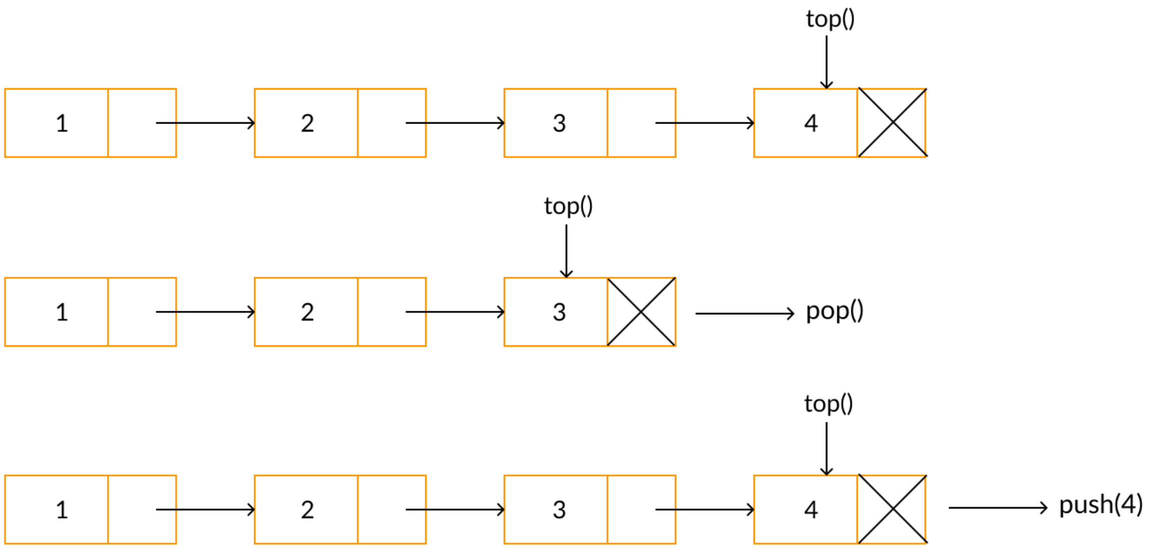


Initially, the topmost element from the array is 5.

Performing a pop operation deletes the topmost element 5 from the array and makes 4 as the topmost element.

Now, on performing push operation to add element 5, the element is added and is made as the topmost element.

## 2. Stacks using Linked Lists



Initially, the topmost element in the linked list is 4.

Performing a pop operation deletes the topmost element 4 from the list and makes 3 as the topmost element.

Now, on performing push operation to add element 4, the element is added and is made as the topmost element.

**Stack Operations**

1. **push()** - Adds an element at the top of the stack. If the stack is full, then it is an overflow.
2. **pop()** - Deletes an element from the top of the stack. If the stack is empty, then it is an underflow.
3. **peek()** - Displays the topmost element in the stack.
4. **isEmpty():** Returns true if stack is empty, else false.