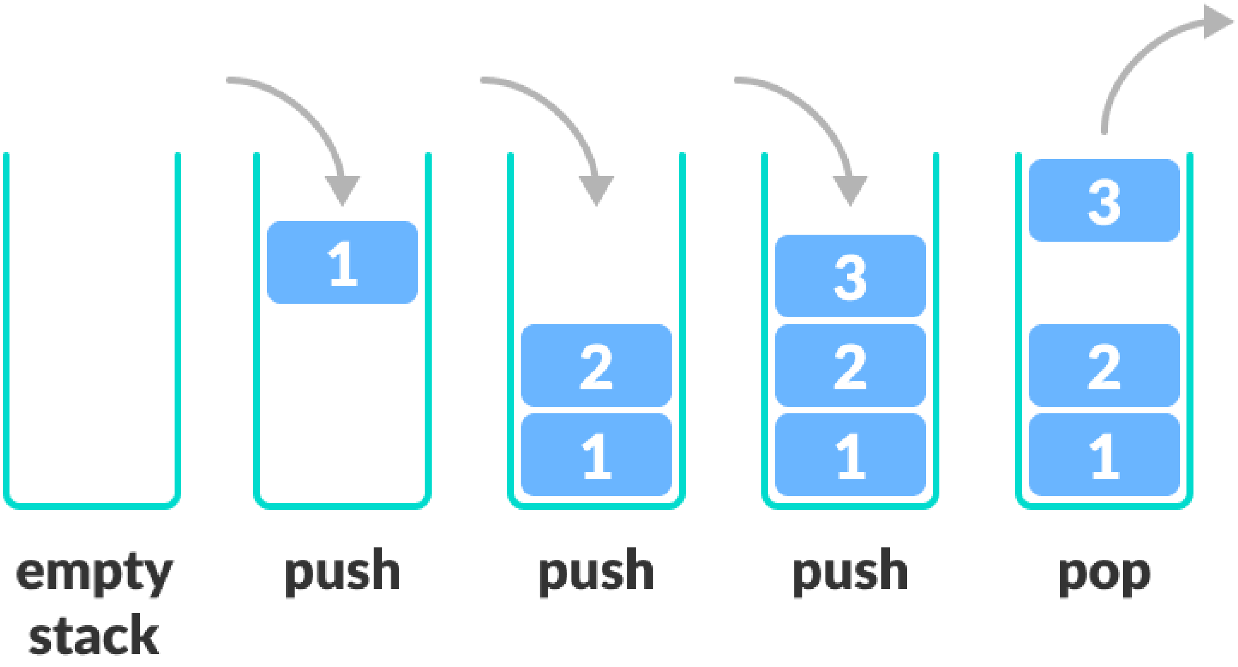
Stacks

Introduction

- Stacks are **linear** data structures.
- They are **LIFO** (Last In First Out) or **FILO** (First In Last Out) data structures.
- Operations can be performed **only on the top element** (or the last inserted element) of the stack.



Operations on stacks

There are specific basic operations that can be performed on a stack that allows us to work with them

- **Push** - Addition of a new element on top of the stack
- **Pop** - Removing the top element of the stack
- **Peek** - Read from the top element of the stack without removing
- **isEmpty** - Check if the stack is empty

Depending on the programming languages, there might also be predefined functions that allow operations like checking if finding the size of the stack etc.

Implementing Stacks

There are multiple ways of implementing a stack in whichever language you prefer coding in. Stacks can be implemented using arrays or linked lists.

https://www.geeksforgeeks.org/stack-data-structure-introduction-program/ : basic implementation of stacks

Some programming language built in implementations of stacks. Being familiar with these will be useful for a quick implementation especially when using them as part of a bigger solution. (Few of these applications are listed in the Additional Reads section)

Refer to the links below to learn about predefined implementations stacks in different languages

- C++: **https://www.geeksforgeeks.org/stack-in-cpp-stl/**
- Java: **https://www.geeksforgeeks.org/stack-class-in-java/**
- Python: **https://www.geeksforgeeks.org/stack-in-python/**

Refer to the links given below to read about some standard operations performed on stacks:

- delete middle element of a stack: **https://www.geeksforgeeks.org/delete-middle-element-stack/**
- Sort a stack: **https://www.geeksforgeeks.org/sort-stack-using-temporary-stack/**
- Reverse a stack: **https://www.geeksforgeeks.org/reverse-stack-without-using-extra-space/**
- Sorting array using stacks: **https://www.geeksforgeeks.org/sorting-array-using-stacks/**

Application of Stacks

1. Infix to prefix/postfix expression: **https://www.geeksforgeeks.org/stack-set-2-infix-to-postfix/**
2. Tower of Hanoi: **https://www.geeksforgeeks.org/iterative-tower-of-hanoi/**
3. Tree traversals: **https://www.geeksforgeeks.org/tree-traversals-inorder-preorder-and-postorder/**
4. Stock span problem: **https://www.geeksforgeeks.org/the-stock-span-problem/**