

# Stack – Introduction

## 1. Introduction

A **Stack** is a **linear data structure** that follows a specific order for performing operations.

The order followed by a stack is called **LIFO — Last In, First Out**.

This means:

- The element added last is removed first
- The element added first is removed last

Stacks are widely used in **program execution, memory management, and algorithm design**.

---

## 2. What is a Stack?

A stack is a collection of elements where:

- Insertion happens at one end only
- Deletion also happens at the same end

This end is called the **Top** of the stack.

There is **no direct access** to middle elements.

---

## 3. Real-Life Example of Stack

A simple real-life example of a stack is:

- A stack of plates
- Books stacked one over another
- Undo/Redo operations in applications

You can only:

- Add a plate on top

- Remove the top plate
- 

## 4. Stack Principle (LIFO)

**LIFO (Last In, First Out)** means:

- The last element inserted is the first one to be removed

Example:

```
Push 10  
Push 20  
Push 30
```

Stack:

```
Top → 30  
  20  
  10
```

Pop:

```
30 is removed first
```

---

## 5. Basic Stack Operations

A stack mainly supports these operations:

1. **Push** – Add an element to the top
  2. **Pop** – Remove the top element
  3. **Peek / Top** – View the top element
  4. **isEmpty** – Check if stack is empty
  5. **isFull** – Check if stack is full (array stack)
- 

## 6. How Stack Works (Plain English Logic)

1. Stack starts empty
  2. When an element is added, it goes to the top
  3. When an element is removed, it is removed from the top
  4. The top pointer updates after every operation
  5. Stack never allows random access
- 

## 7. Stack Representation

Stacks can be implemented using:

- **Array**
- **Linked List**

Both implementations follow the same LIFO rule.

---

## 8. Stack Overflow and Underflow

### Stack Overflow

- Occurs when trying to push an element into a full stack

### Stack Underflow

- Occurs when trying to pop an element from an empty stack

These are common stack errors.

---

## 9. Applications of Stack

Stacks are used in:

- Function calls (call stack)
- Expression evaluation
- Undo/Redo operations
- Syntax checking

- Backtracking algorithms
  - Browser history navigation
- 

## 10. Advantages of Stack

- Simple to implement
  - Efficient for LIFO operations
  - Useful for recursive problems
  - Helps manage function calls
- 

## 11. Limitations of Stack

- Limited access to elements
  - Fixed size (array implementation)
  - Not suitable when random access is needed
- 

## 12. Time Complexity of Stack Operations

Operation	Time Complexity
Push	O(1)
Pop	O(1)
Peek	O(1)

All stack operations are very fast.

---

## 13. Summary

- Stack is a linear data structure
- Follows LIFO principle
- Operations happen at the top
- Used widely in real-world systems

- Simple yet powerful structure
-