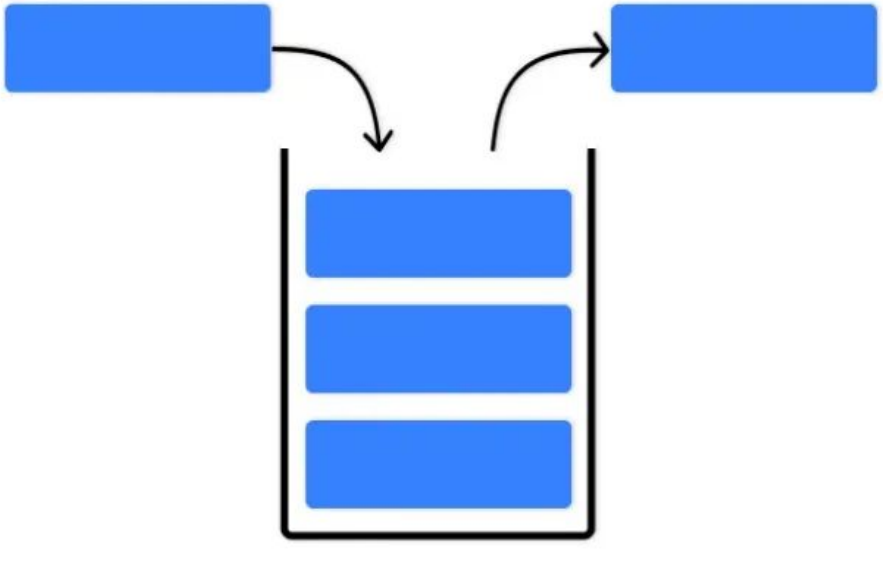
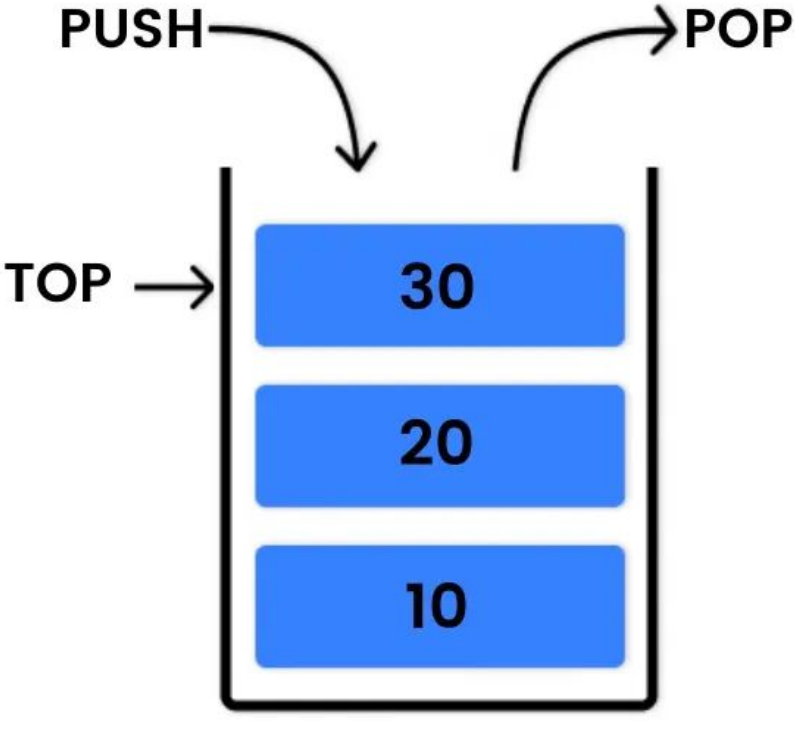
Stack
Data Structure in Python



A stack is a linear data structure that stores the elements in Last In First Out (LIFO) manner. In stack, addition of new element (PUSH) or removal (POP) of an element always take place at the same end called TOP.



For an empty stack value of TOP is -1. When initializing the stack, we set its value to -1 so that we can check if the stack is empty by comparing TOP == -1.

Operations of Stack

1. Push operation: It add a new element to the top of a stack. In push operation before pushing a new element, first we need to check if the stack is already full.

2. Pop: Remove an element from the top of a stack. In pop operation before popping an element, first we need to check if the stack is already empty.

3. Peek or Top: Return the value of the top element.

4. isEmpty: Checks whether the stack is empty.

5. isFull: Checks whether the stack is Full.

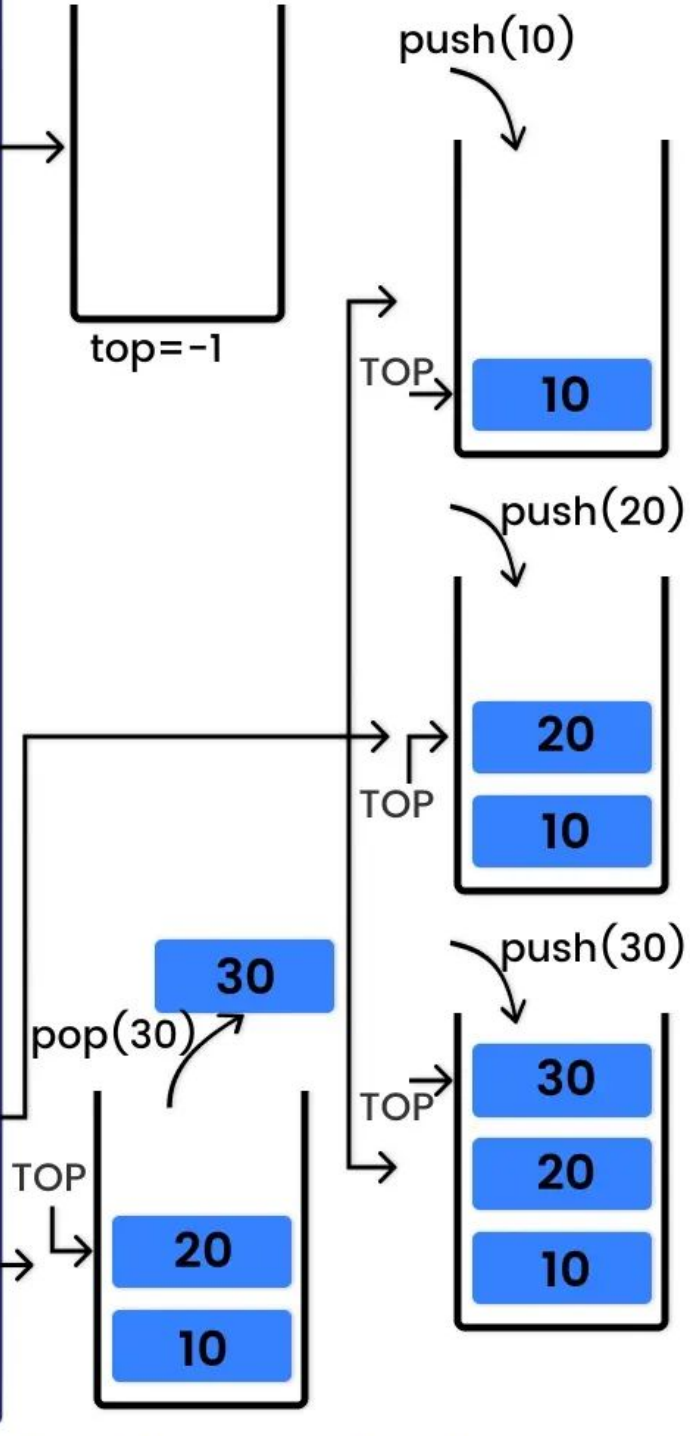
Stack Overflow and Underflow

Overflow: If stack is Full, then the stack is said to be Overflow. In other words, stack overflow occurs when we try to add or push the new element beyond its maximum allocated size.

Underflow: If stack is empty then it is said to be Underflow. In other words, stack underflow occurs when we try to remove or pop an element from the stack when there are no elements in the stack or TOP==-1.

Implementation of stack

```
class Stack:
    def __init__(self, size):
        # initializing stack
        self.stack = []
        self.maxSize = size
        self.top = -1
    def push(self, x):
        if self.top == self.maxSize-1:
            print('Stack Overflow')
        else:
            self.x = x
            self.stack.append(x)
            self.top += 1
    def pop(self):
        if self.top == -1:
            print('Stack Underflow')
        else:
            self.stack.pop()
            self.top -= 1
    def isEmpty(self):
        print(self.top == -1)
    def isFull(self):
        print(self.top == self.maxSize-1)
    def peek(self):
        print(self.stack[-1])
    def displayElements(self):
        for i in range(self.top, -1, -1):
                print(self.stack[i])
stackObj = Stack(3)
stackObj.push(10)
stackObj.push(20)
stackObj.push(30)
stackObj.displayElements()
stackObj.isFull()
stackObj.pop()
stackObj.displayElements()
stackObj.isEmpty()
stackObj.peek()
```



Output:

```
30
20
10
True
20
10
False
20
```

Save this post if you found it helpful.

THANK YOU!