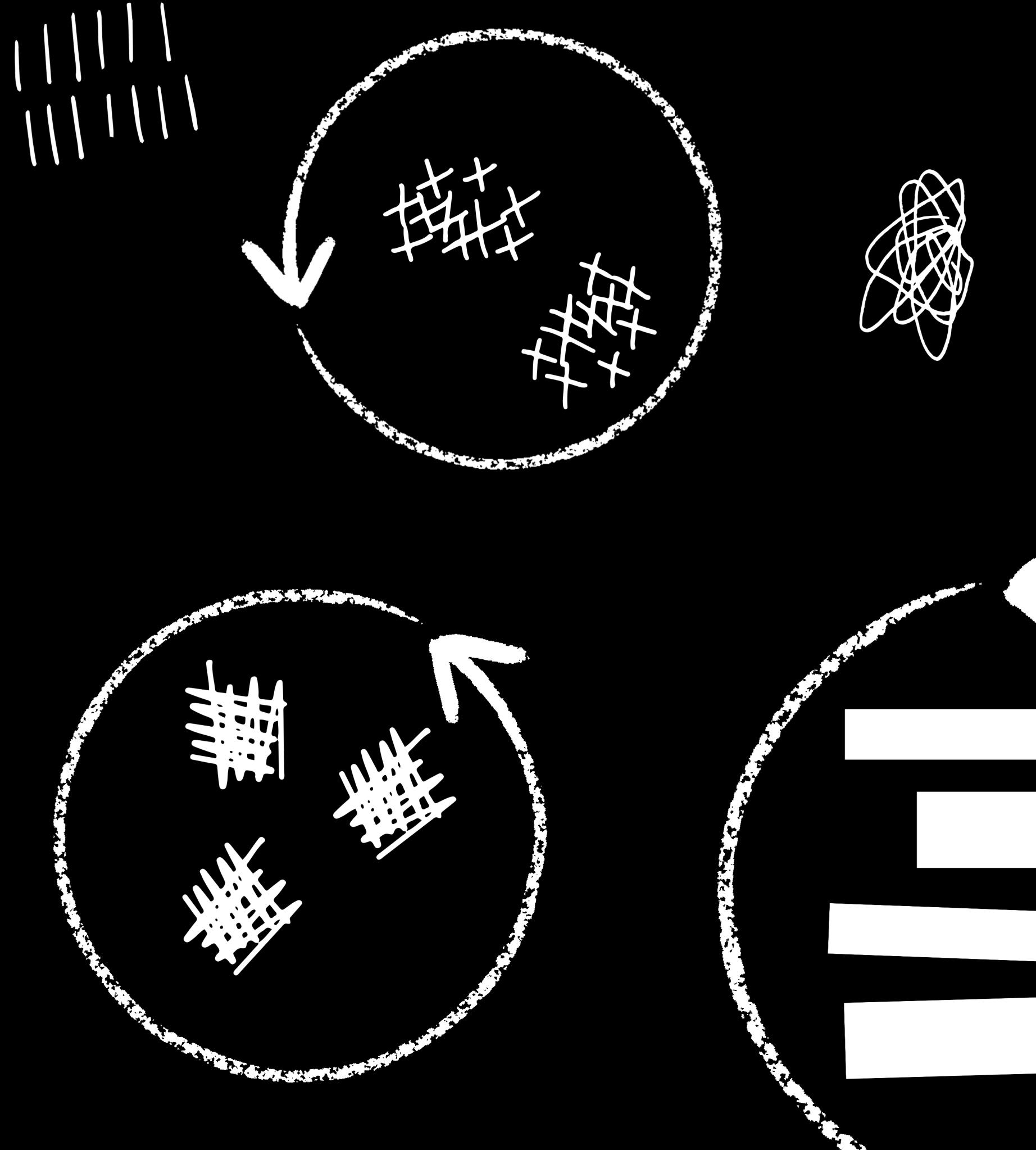


# DSA Session

A clear step-by-step process for  
solving Stack.

# Content :

- What is stack.
- Methods used in stack.
- Approach.
- Practical working .
- Programming Demo.



# What is Stack ?

A stack is a linear data structure that stores items in a Last-In/First-Out (LIFO) or First-In/Last-Out (FILO) manner. In stack, a new element is added at one end and an element is removed from that end only. The insert and delete operations are often called push and pop.



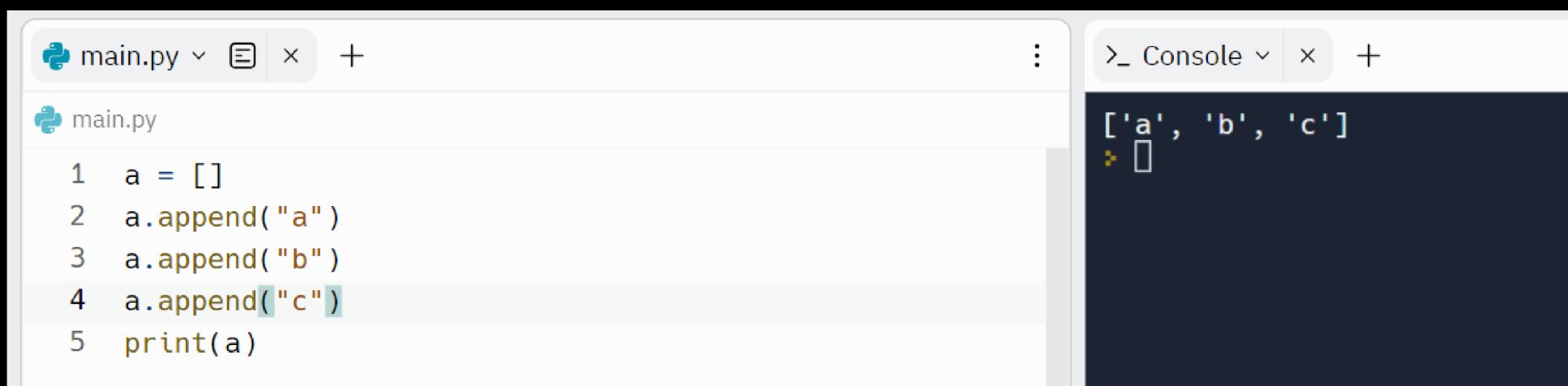
# Methods

- `empty()` – Returns whether the stack is empty – Time Complexity:  $O(1)$
- `size()` – Returns the size of the stack – Time Complexity:  $O(1)$
- `top() / peek()` – Returns a reference to the topmost element of the stack – Time Complexity:  $O(1)$
- `push(a)` – Inserts the element ‘a’ at the top of the stack – Time Complexity:  $O(1)$
- `pop()` – Deletes the topmost element of the stack – Time Complexity:  $O(1)$

# Lifo Last in first out

In stack, a new element is added at one end  
and an element is removed from that end

only.  
↓  
↓  
↓



The image shows a screenshot of a Python development environment. On the left, there is a code editor window titled "main.py" containing the following Python code:

```
1 a = []
2 a.append("a")
3 a.append("b")
4 a.append("c")
5 print(a)
```

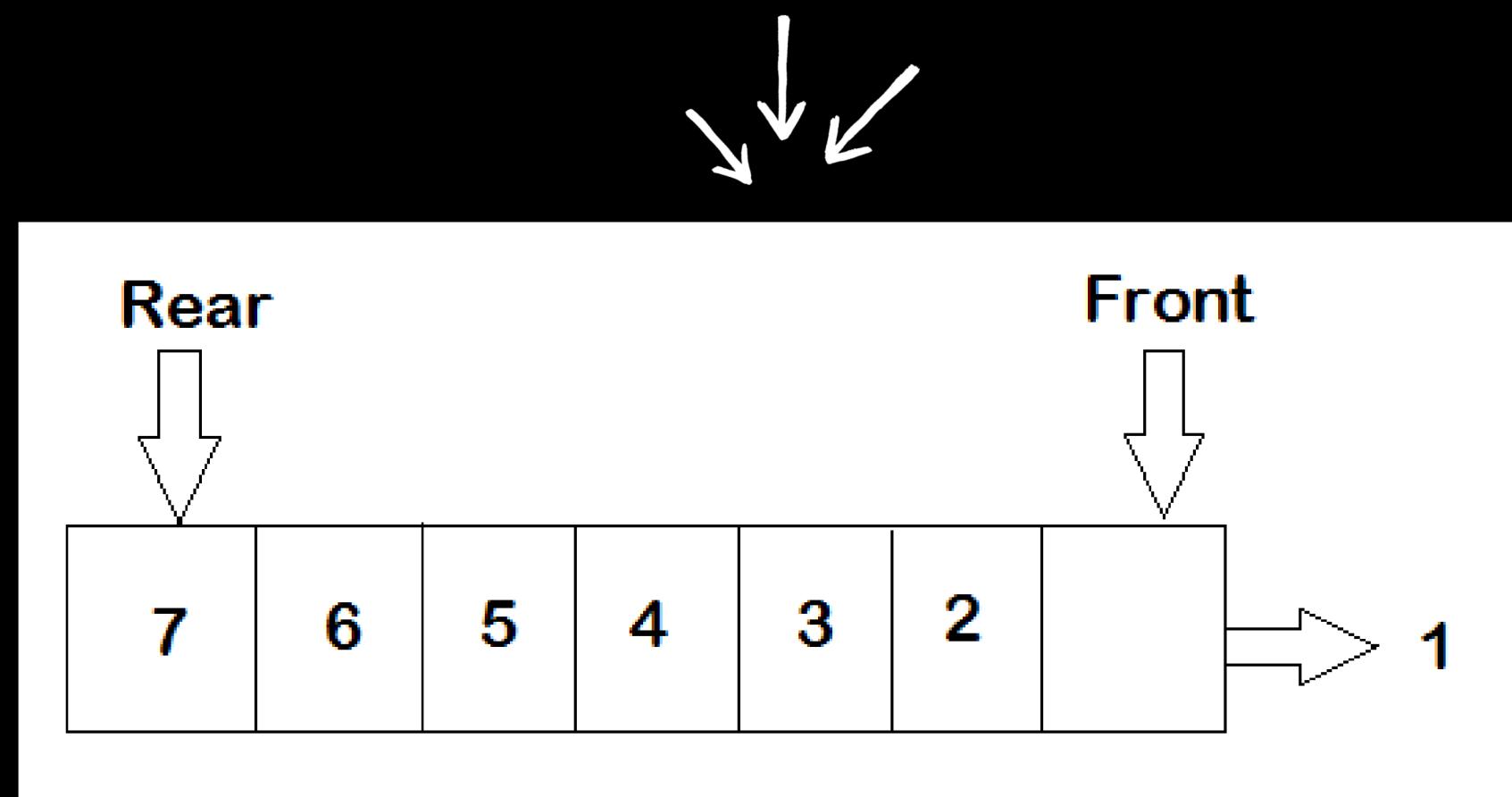
On the right, there is a terminal window titled "Console" showing the output of the code:

```
['a', 'b', 'c']
```

# Fifo first in first out

In, First Out (FIFO) principle. It means that the first added element gets removed first from the list.

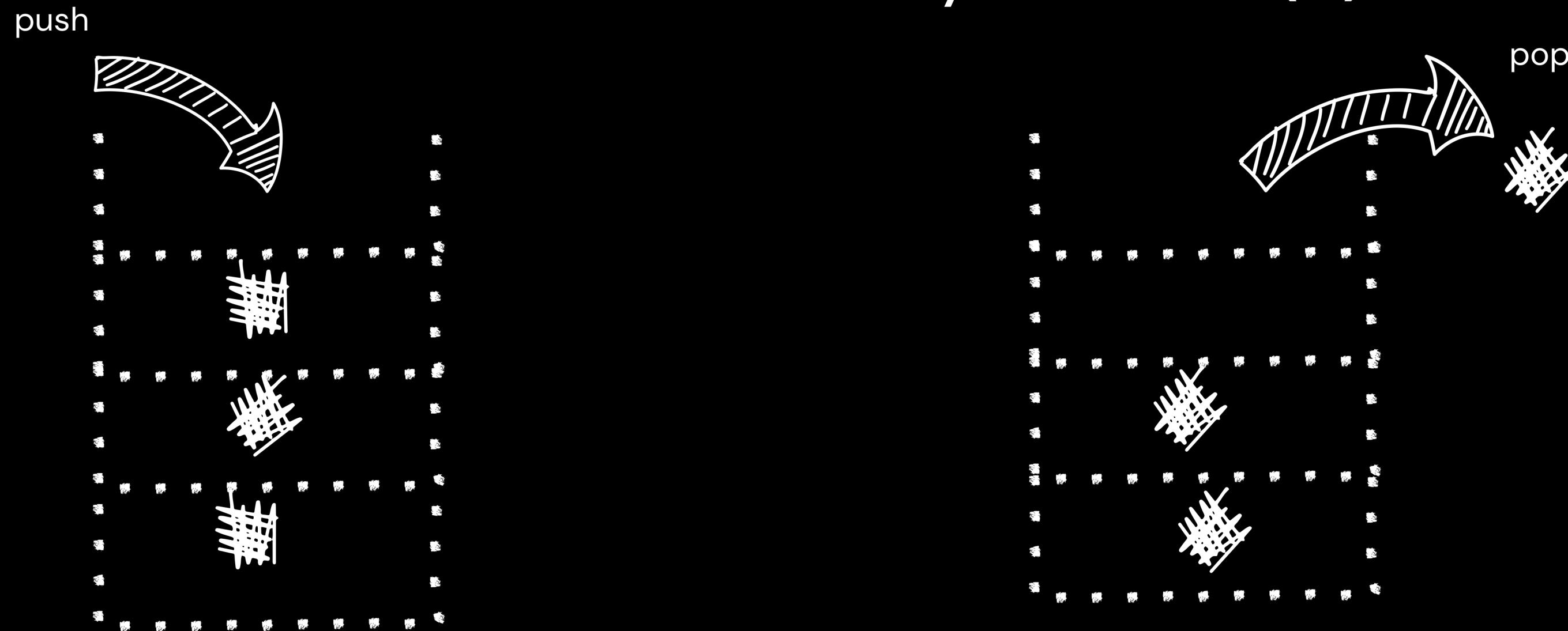
Pointers. It has only one pointer- the top. This pointer indicates the address of the topmost element or the last inserted one of the stack.

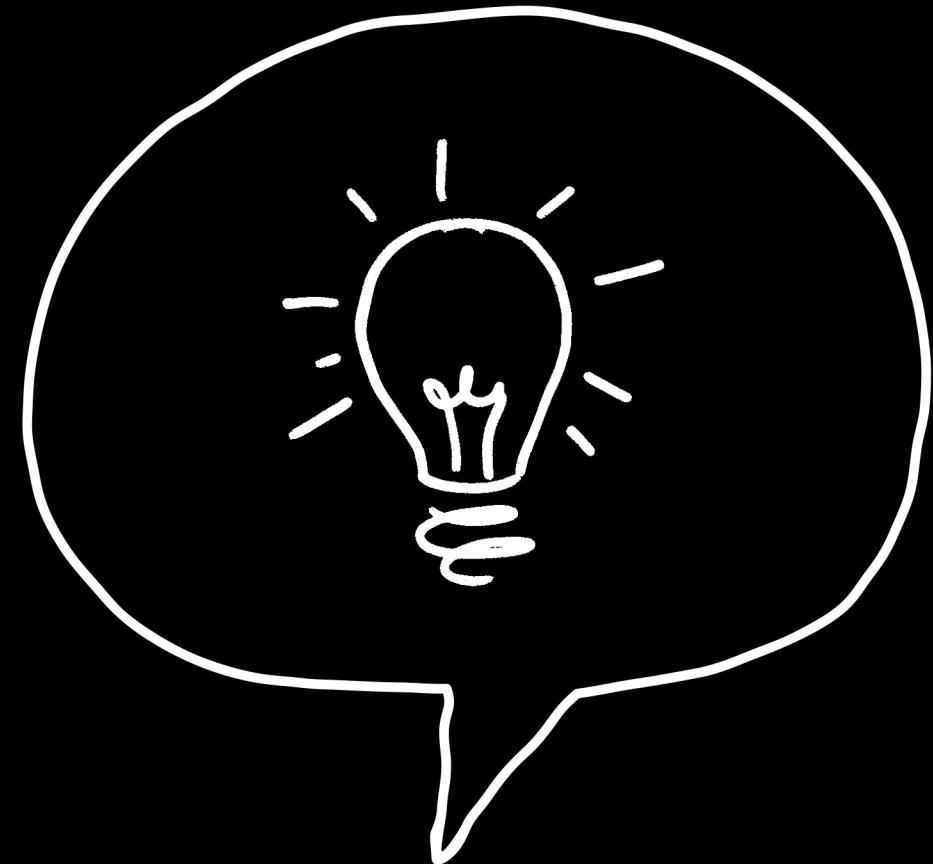


# Working :

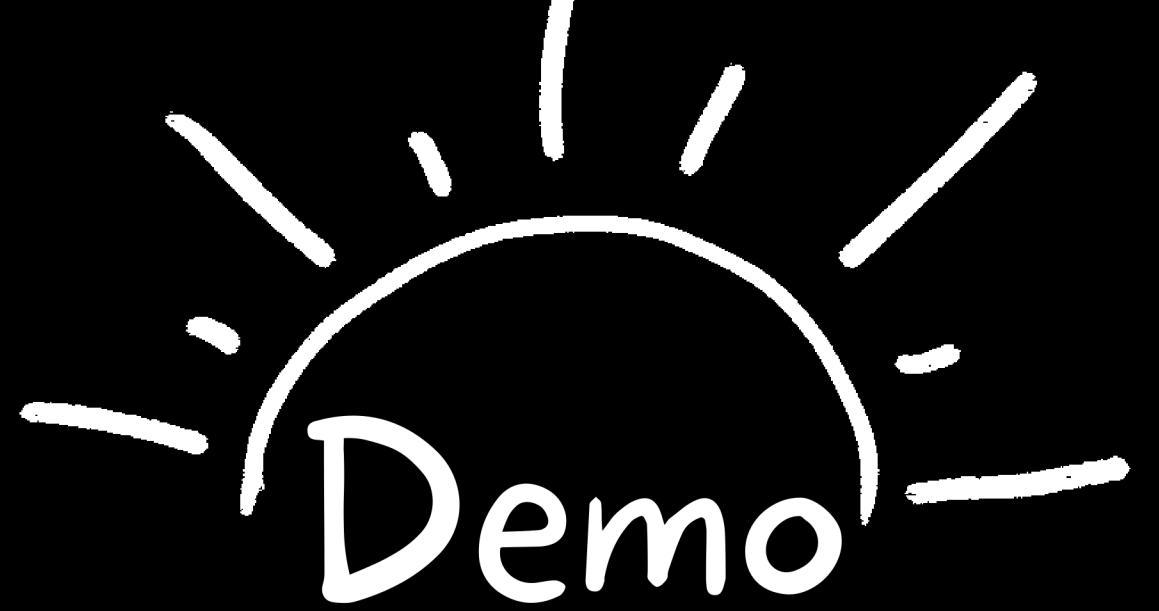
push and pop demo..  $O(1)$

for search element by value : $O(n)$





Time for  
Programming  
Example



```
main.py ✘ x + : _ Console ✘ x +
```

main.py

```
1 # Python program to
2 # demonstrate stack implementation
3 # using list
4
5 stack = []
6
7 # append( ) function to push
8 # element in the stack
9 stack.append('a')
10 stack.append('b')
11 stack.append('c')
12
13 print('Initial stack')
14 print(stack)
15
16 # pop( ) function to pop
17 # element from stack in
18 # LIFO order
19 print('\nElements popped from stack:')
20 print(stack.pop())
21 print(stack.pop())
```

Initial stack  
['a', 'b', 'c']

Elements popped from stack:  
c  
b  
a

Stack after elements are popped:  
[]  
:> []



Any Difficulty?

