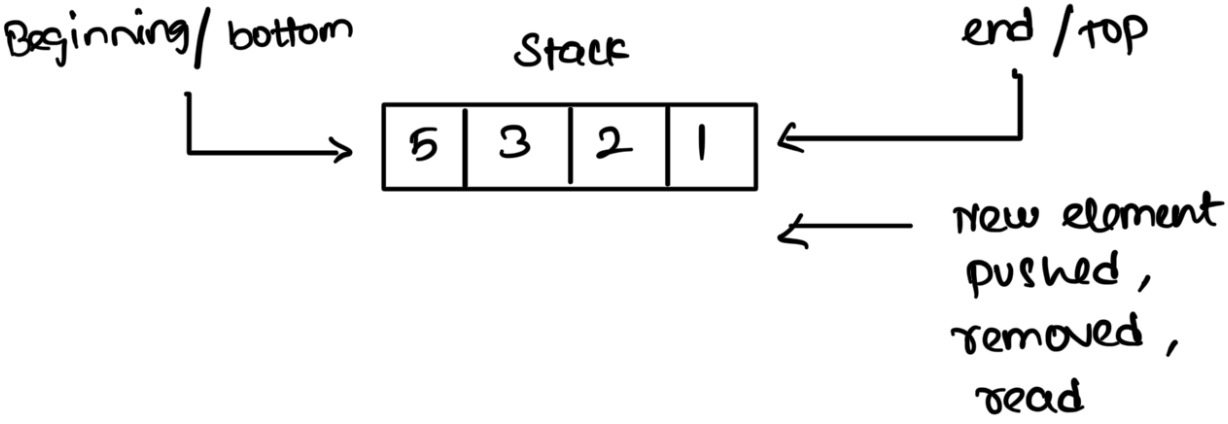
Stack and Queue

- Good for handling temporary data.
- Temporary data is information that doesn't have any meaning after it's processed., so no need to keep it.
- However, the order is which it is processed the data is important.
- Stack and Queue allow to handle data in order, and then get rid of it once it is not needed anymore.

STACK :

- Stack stores data in the same way that array do, its a simple list of elements.
- It has three constraints
    1). Data can be inserted at the end / top of the stack.
    ii). Data can be read from the end / top of the stack.
    iii). Data can be removed from the end / top of the stack.

- end of stack refered as top, beginning of stack referred as bottom.



- Inserting new element into the stack also known as pushing into stack.
- All new element is inserted at the top / end of the stack.
- Removing an element from the stack is known as popping from stack.

- element can be removed from the top / end of the stack.

- Data can be read from top/ end of the stack.
- Stack operation is LIFO - Last In, First out.

In Action :

eg: Linter program:

- Identify opening and closing braces.

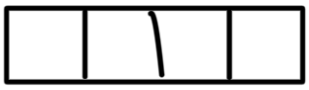
```
// TODO
```

QUEUE :

- A queue also deals with temporary data.
- Similar to Stack, difference lies in order.
- Stores data in array.
- FIFO - First In, First Out.

- Three restriction queue has
  1). Data can be inserted at the end / top of the Queue.
  11). Data can be read only from the front / bottom of the queue.
  111). Data can be removed from the front / bottom of the queue.

front/bottom top/end



← Data enqueue

← Data read, removed

eg:

```
// TODO
```