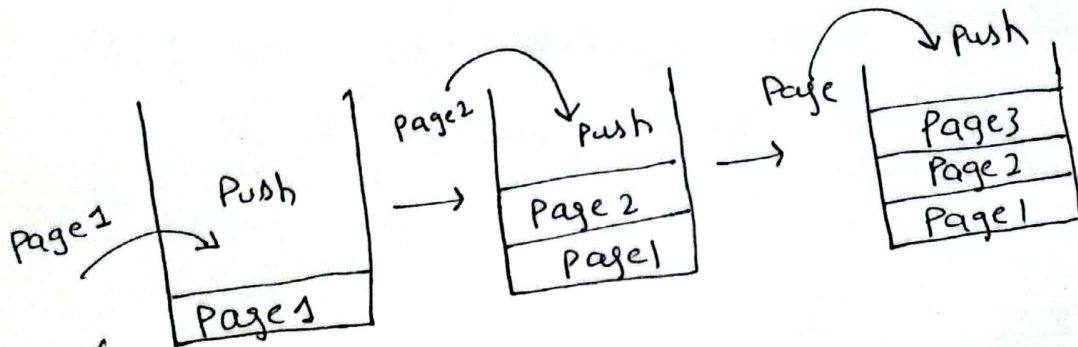


## ⑦ Stack

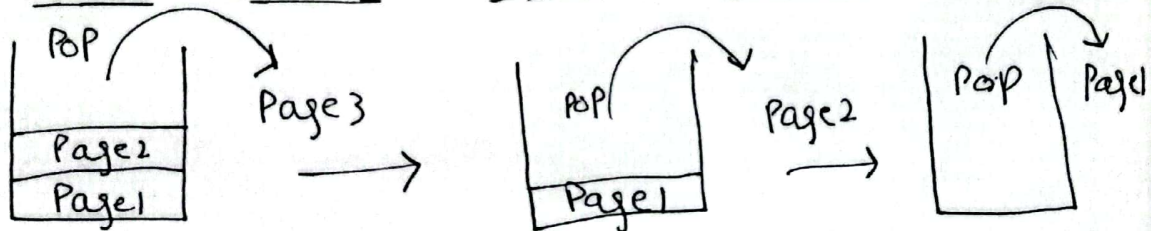
### Application :

If you visit page 1 then go to page 2  
& then page 3 in a browser.  
↳ Now you are at page 3  
and you click on back button.  
⇒ so using stack this implementation  
is easy.



↳ So basically in push means you put  
every page in the stack.

NOW click back button



↳ So POP means you throw current  
entry of stack and got last  
entry you visited.

⇒ This is last in First out operation.

↳ LIFO

↳ so stack is LIFO Data  
structure.

Push/Pop element  $\Rightarrow O(1)$  (T.C)  
search element by value  $\Rightarrow O(n)$  (T.C)  
↳ because you have to go through every element.

### Use of Stack:

→ Function calling in any programming language is managed using a stack.

↳ If function a was call the it call function b & then b called c.  
⇒ Compiler store the order so that it can return to the previous function.

→ ~~Ctrl~~ Ctrl + Z function in any editor uses stack to track last operation.

### Stack Implementation:

Python : using list

C++ : std::stack

↳ 

```
std::stack<int> stk;  
stk.push(5);  
stk.push(89);  
stk.pop();
```

→ In Python if we use list to implement stack it can be costly as lists are dynamic so (if space ended) it has to copy all the elements to new space.

→ So we will use collection.deque() for stack - These are based on doubly linked lists.