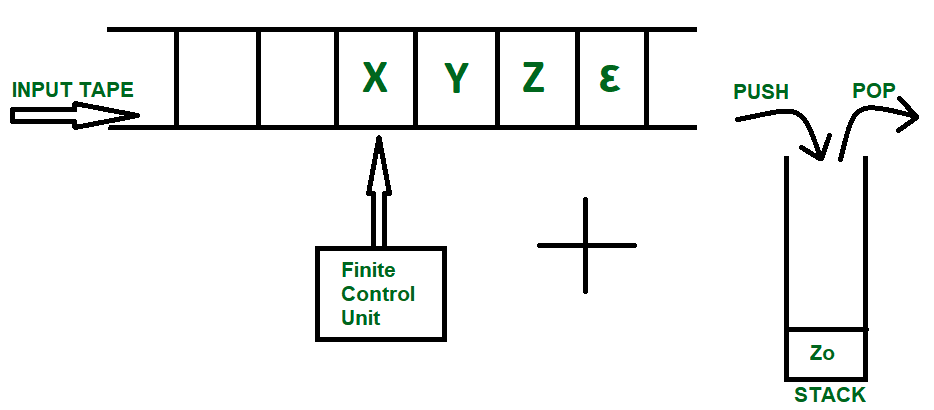
Pushdown Automata is a finite automata with extra memory called stack which helps Pushdown automata to recognize Context Free Languages.  
   
A Pushdown Automata (PDA) can be defined as :

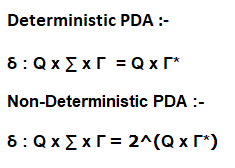
* Q is the set of states
* ∑is the set of input symbols
* Γ is the set of pushdown symbols (which can be pushed and popped from stack)
* q0 is the initial state
* Z is the initial pushdown symbol (which is initially present in stack)
* F is the set of final states
* δ is a transition function which maps Q x {Σ ∪ ∈} x Γ into Q x Γ\*. In a given state, PDA will read input symbol and stack symbol (top of the stack) and move to a new state and change the symbol of stack.



The diagram above shows a input tape which is how a [Finite Automata](https://www.geeksforgeeks.org/toc-finite-automata-introduction/) works, the strings are accepted into the tape and the read header keeps getting updated according the instructions provided by Finite Control Unit. Pushdown Automata on the other hand is a combination of this tape and a Stack data structure.

**This assumption helps us in two ways :-**

1. We overcome the underflow condition thus saving any memory to keep a check on Stack empty.
2. Initial Stack symbol can be used to declare that string processing has been done successfully.

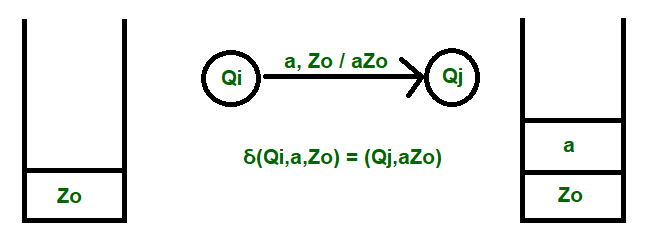


**Tau Symbol (Γ)** is used to denote all the Stack Alphabets. Each input alphabet ( same or different ) can be denoted by a different Stack symbol. It’s also necessary as it conveys the topmost element of the stack to the machine.

**Delta Function (δ**)**)** is the transition function, the use of which will become more clear by taking a closer look at the Three Major operations done on Stack :-

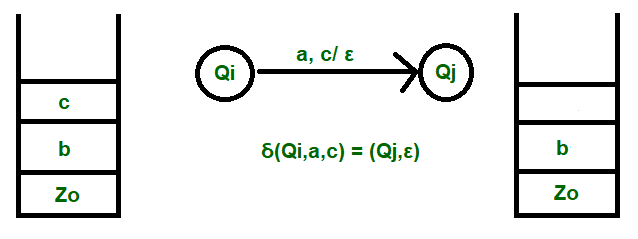
1. Push
2. Pop
3. Skip

**1.PUSH**



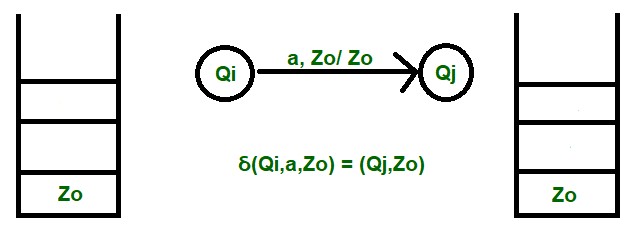
Push Operation is done as shown in the diagram.  
**The transition takes place in the order :-***Input, Topmost Element / Final List*  
Here, a is the input element, which is inserted into the stack, thus making the final content to be aZo.

**2.POP**



Pop Operation is done as shown in the diagram.  
**The transition takes place in the order :-** *Input Element, Topmost Element / Removal Confirmation*  
Here, a is the input, c is the element to be deleted and the removal confirmation is shown by Epsilon symbol declaring that the immediate has been popped and is Empty.

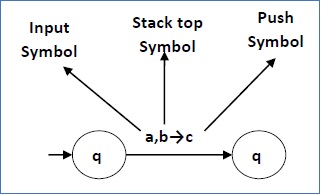
**3.SKIP**



Skip operation is done as shown in the diagram.  
**The transition takes place in the order:-***Input Element, Topmost element/Topmost Element*

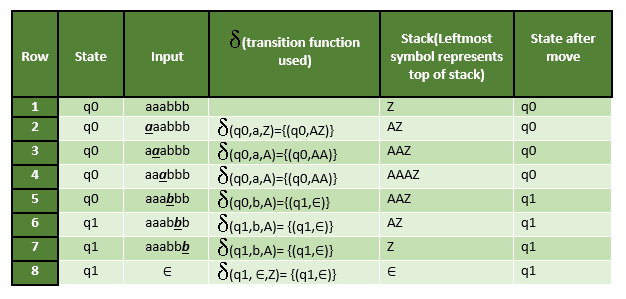
Here, a is the input and the stack remains unchanged after this operation.

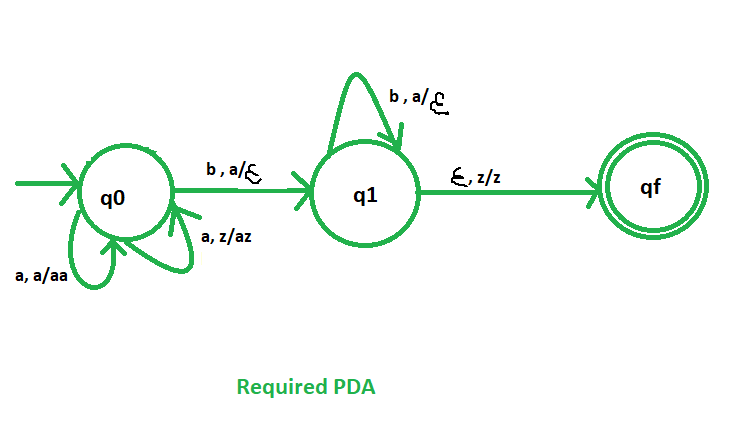
Hence, this concludes the detailed study on how a Pushdown Automata works. Next we will be heading onto some examples to make working and transitions more clear

  
This means at state **q1**, if we encounter an input string **‘a’** and top symbol of the stack is **‘b’**, then we pop **‘b’**, push **‘c’** on top of the stack and move to state **q2**.

**Turnstile notation**  
⊢ sign is called a “turnstile notation” and represents  
one move.  
⊢\* sign represents a sequence of moves.  
Eg- (p, b, T) ⊢ (q, w, α)  
This implies that while taking a transition from state p to state q, the input symbol ‘b’ is consumed, and the top of the stack ‘T’ is replaced by a new string ‘α’

**Example :** Define the pushdown automata for language {anbn | n > 0}  
**Solution :**M = where Q = { q0, q1 } and Σ = { a, b } and Γ = { A, Z } and &delta is given by :

&delta( q0, a, Z ) = { ( q0, AZ ) }  
&delta( q0, a, A) = { ( q0, AA ) }  
&delta( q0, b, A) = { ( q1, ∈) }  
&delta( q1, b, A) = { ( q1, ∈) }  
&delta( q1, ∈, Z) = { ( q1, ∈) }  
   
Let us see how this automata works for aaabbb.  




Example 2: L= (anbncm, where n>=1,m>=0)

Example 3 Design a PDA for accepting a language {anb2n | n>=1}.

Example 4.