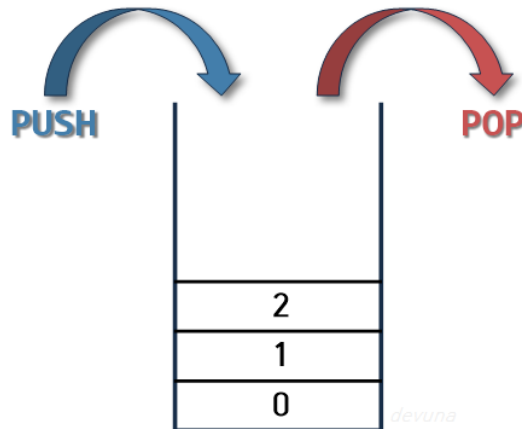


스택과 큐

스택 (Stack) 이란?



LIFO (Last In First Out) 방식

- 스택(Stack) 자료구조는, 책을 쌓는 것처럼 차곡차곡 쌓아 올린 형태의 자료구조를 의미한다.
- 즉, 후입선출(LIFO, Last In First Out) 방식의 자료구조이다.

스택의 특징

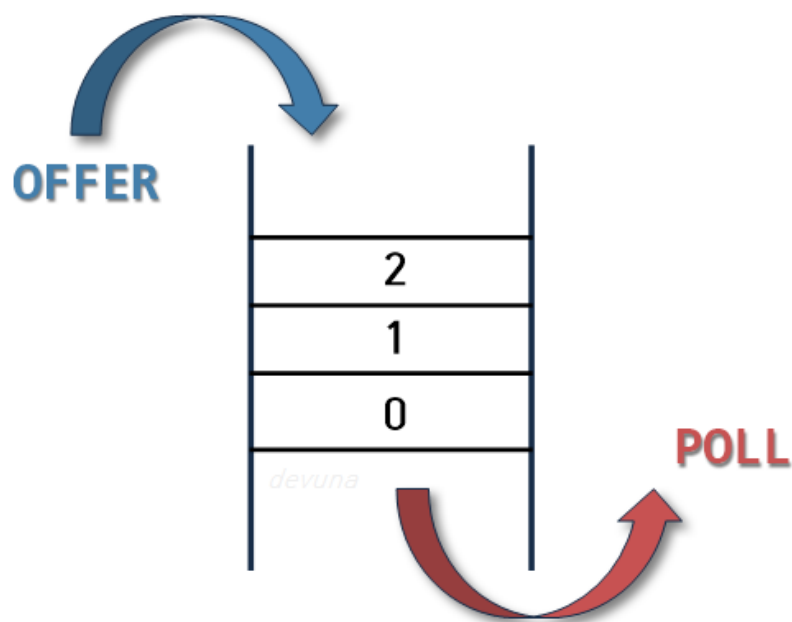
- 스택 내부의 데이터는, top 을 통해서만 접근할 수 있다.
 - top 은 가장 최근(마지막)에 들어온 자료를 의미한다.
- 스택에 데이터를 삽입할 때는, top 위에 쌓게 되며 ('push' 연산)
- 스택에서 데이터를 삭제할 때는, top 에 위치한 데이터를 삭제하게 된다. ('pop' 연산)
- 즉, 스택은 시간 순서에 따라 데이터가 쌓이게 되므로,
- 가장 마지막에 삽입된 데이터가 가장 먼저 삭제된다는 특징을 가지게 된다.
- 이러한 스택의 구조를 후입선출(LIFO, Last-In-First-Out) 구조라고 한다.

스택의 활용 예시

- 웹 브라우저 방문기록 (뒤로 가기)

- 가장 마지막에 열린 페이지부터 다시 보여준다.
- **실행 취소 (undo)**
 - 가장 마지막에 실행된 것부터 실행을 취소한다.
- **역순 문자열 만들기**
 - 가장 마지막에 입력된 문자부터 출력한다.
- **후위 표기법 계산**
- **수식의 괄호 검사**
 - 연산자 우선순위 표현을 위한 괄호 검사

큐 (Queue) 란?



FIFO(First In First Out) 방식

- 큐(Queue) 는 "줄을 서서 기다린다."라는 사전적 의미를 가지고 있다.
- 따라서, 큐 (Queue) 자료구조는, **먼저 들어온게 먼저 나가는 선입선출(FIFO, First In FirstOut) 방식**의 자료구조이다.

큐의 특징

- 스택은, top 을 통해서만 삽입, 삭제가 이루어졌었다.

- 하지만 큐는 삽입, 삭제가 다른 방향에서 이루어진다.
- 이때, 삭제 연산만 수행되는 곳을 프론트(front), 삽입 연산만 수행되는 곳을 리어(rear)라고 한다.
 - 프론트(front)에서 이루어지는 삭제 연산을 디큐(enQueue)라고 하며,
 - 리어(rear)에서 이루어지는 삽입 연산을 인큐(enQueue)라고 한다.
- 즉, 큐는 가장 먼저 삽입된 데이터가 가장 먼저 삭제된다는 특징을 가지게 되며,
- 이러한 큐의 구조를 선입선출(FIFO, First In FirstOut) 구조라고 한다.

큐의 활용 예시

- 큐는 주로, 데이터가 입력된 순서에 따라 처리되어야 할 때 사용된다.
 - 일상생활에서, 줄을 서서 기다려야하는 모든 행동
 - 은행 업무
 - 콜센터 고객 대기시간
 - 놀이동산
 - 프로세스 관리
 - 너비 우선 탐색(BFS, Breadth-First Search) 구현
 - 캐시(Cache) 구현