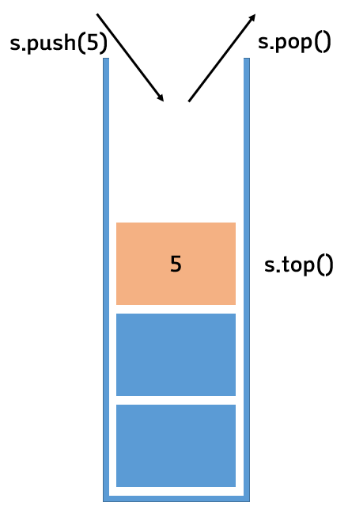
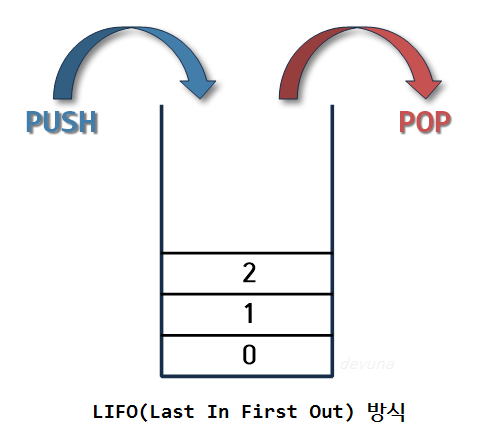
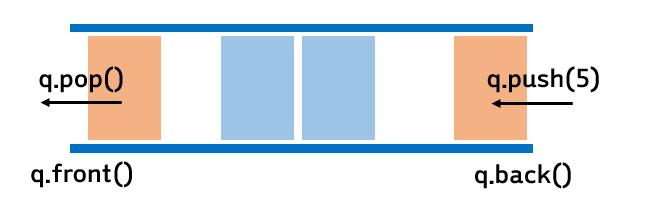
**Stack & Queue**

1. **Stack**
   1. 스택의 개념
      1. 스택(Stack)이란 쌓아 올린다는 것. 즉, 차곡차곡 쌓아 올린 형태의 자료구조
   2. 스택의 특징
      1. 같은 구조와 크기의 자료를 정해진 방향으로만 쌓을수 있다.
      2. ‘push’를 이용해 top에 삽입하고, ‘pop’을 이용해 top을 삭제한다.
      3. 후입선출(LIFO, Last-in-First-out) 가장 마지막에 삽입된 자료가 가장 먼저 삭제
      4. 빈 스택에서 원소 추출 : Stack Underflow, 스택이 넘치는 경우 : Stack overflow
   3. 스택의 활용 예시
      1. 웹 브라우저 방문기록(뒤로가기) : 가장 나중에 열린 페이지부터 다시 보여준다.
      2. 역순 문자열 만들기 : 가장 나중에 입력된 문자부터 출력
      3. 실행 취소, 후위 표기법 계산, 수식의 괄호 검사(연산자 우선순위 표현)
   4. 시간 복잡도
      1. 삽입/삭제 : 원소를 삽입/삭제하는 경우 : O(1)
   5. 장점 / 단점
      1. 데이터의 삽입과 삭제가 빠르다.
      2. 탐색을 하려면 원소를 하나하나 옮겨가며 해야함, 맨 위 원소만 접근 가능
2. **Queue**
   1. 큐의 개념
      1. 큐는 사전적으로 줄, 혹은 줄을 서서 기다리는 것을 의미
      2. 놀이동산에서 줄을 서서 기다리는 것, 은행에서 먼저 온 사람의 업무를 처리
   2. 큐의 특징
      1. 선입선출(FIFO, First-in-First-out) : 먼저 넣은 데이터가 먼저 나오는 구조
      2. ‘push’ 되는 곳을 ‘back(rear)’, ‘pop’되는 곳을 ‘front’라고 한다.
      3. 접근방법은 front와 back으로 가능
      4. 가장 먼저 들어온 원소가 가장 먼저 삭제된다.
   3. 큐의 활용 예시
      1. 큐는 주로 데이터가 입력된 시간 순서대로 처리해야 할 필요가 있는 상황
      2. 은행 업무, 콜센터 고객 대기시간, 프로세스 관리
      3. 너비우선탐색(BFS, Breath-First Search)구현, 캐시(Cache) 구현
   4. 시간 복잡도
      1. 삽입/삭제 : 원소를 삽입/삭제하는 경우 : O(1)
   5. 장점/단점
      1. 데이터의 삽입/삭제가 빠르다.
      2. Queue의 중간에 위치한 데이터로의 접근이 어렵다.
      3. 선형 큐 : Front는 고정, Back을 이동하면서 데이터를 삭제하는 경우  
         - 데이터를 제거한 경우, 나머지 데이터를 한 칸씩 다 옮겨야한다.
      4. 둘 다 이동하면서 삽입, 삭제를 할 경우  
         - 배열의 끝에 저장되어 있는 상황, Back을 더 이상 이동시킬 수 없어 overflow
      5. 순환 큐(환형 큐) : 선형 큐를 보완하기 위한 방식, front가 큐의 끝에 닿으면  
         큐의 맨 앞으로 자료를 보내서 원형으로 연결.