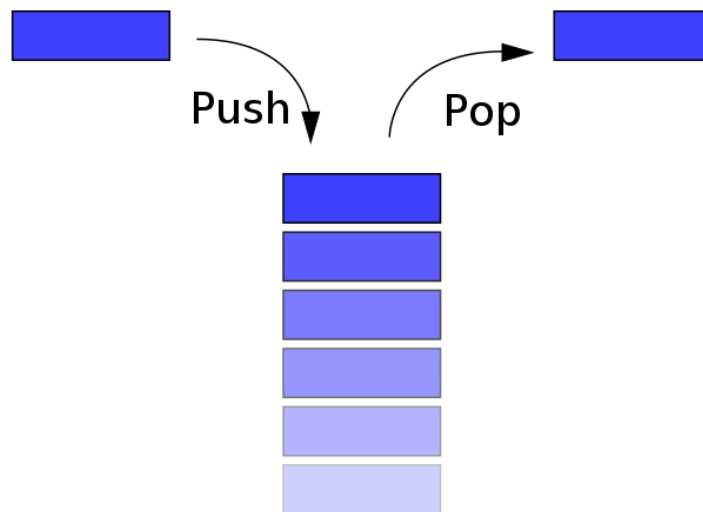


스택, 큐, 덱

위승빈 (wsb0722)

1. 스택(stack)

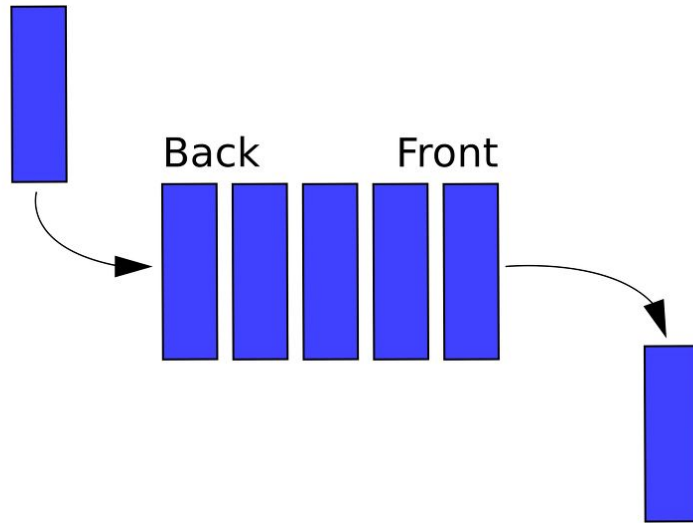
스택이란?: 한쪽 끝에서만 자료를 넣고 뺄 수 있는 자료구조. 즉, 스택과 반대되는 개념



- **LIFO(Last In First Out)**의 구조 형태를 지니고 있다.
- 마지막에 넣은 자료가 의미 있는 문제에 스택이 쓰인다.
- 스택에는 아래와 같은 연산이 있다.
 - push : 스택에 자료를 넣는 연산
 - pop : 스택에서 자료를 빼는 연산
 - top : 스택의 가장 위에 있는 자료를 보는 연산
 - empty : 스택이 비어있는지 아닌지를 알아보는 연산
 - size : 스택에 저장되어 있는 자료의 개수를 알아보는 연산
- 배열이나 리스트를 통해 구현할 수 있다.
- 라이브러리에 구현되어 있는 스택을 쓸 수 있다.
 - Python에서는 스택이 따로 없고 리스트를 사용해 스택처럼 활용할 수 있다.

2. 큐(queue)

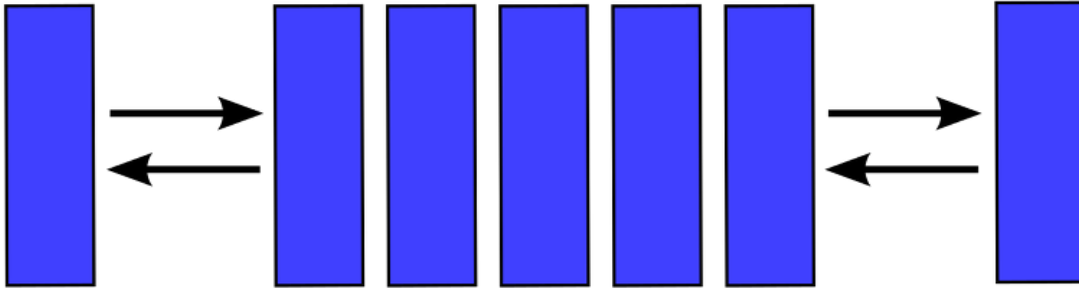
큐란? : 한쪽 끝에서만 자료를 넣고 다른 한쪽 끝에서만 뺄 수 있는 자료구조. 즉, 큐와 반대되는 개념



- **FIFO(First In First Out)**의 구조 형태를 지니고 있다.
- 은행 업무 시스템과 비슷하다. (먼저 번호표를 뽑은 사람이 먼저 업무를 볼 수 있다.)
- 큐에는 아래와 같은 연산이 있다.
 - push : 큐에 자료를 넣는 연산
 - pop : 큐에서 자료를 빼는 연산
 - front : 큐의 가장 앞에 있는 자료를 보는 연산
 - back : 큐의 가장 뒤에 있는 자료를 보는 연산
 - empty : 큐가 비어있는지 아닌지를 알아보는 연산
 - size : 큐에 저장되어 있는 자료의 개수를 알아보는 연산
- 배열이나 리스트를 통해 구현할 수 있다.
- 라이브러리에 구현되어 있는 큐를 쓸 수 있다.
 - Python에서는 큐가 따로 없고 collections 모듈의 deque를 사용해 큐처럼 활용할 수 있다.

3. 덱(deque) : Double-ended queue

덱이란? : 양 끝에서만 자료를 넣고 양 끝에서 뺄 수 있는 자료구조. 즉, 스택과 큐를 합친 형태



- 덱에는 아래와 같은 연산이 있다.
 - push_front : 덱의 앞에 자료를 넣는 연산
 - push_back : 덱의 뒤에 자료를 넣는 연산
 - pop_front : 덱의 앞에서 자료를 빼는 연산
 - pop_back : 덱의 뒤에서 자료를 빼는 연산
 - front : 덱의 가장 앞에 있는 자료를 보는 연산
 - back : 덱의 가장 뒤에 있는 자료를 보는 연산
- 배열이나 링크드 리스트를 통해 구현할 수 있다.
- 라이브러리에 구현되어 있는 덱을 쓸 수 있다.