



# 0x06 큐

☼ 상태	완료
👤 담당자	서연

## 큐란? Queue



한쪽 끝에서 원소를 넣고 한 쪽 끝에서 원소를 빼는 선입선출되는 자료구조임.  
FIFO(First In First Out)

### 성질

1. 큐는 원소의 삽입과 삭제 시 시간복잡도  $O(1)$ 을 가짐.
2. 앞과 뒤 원소 확인에  $O(1)$ 의 시간 복잡도를 가짐.

### 구현

#### 필요한 변수

1. 앞과 뒤를 가리킬 레퍼런스 변수 head, tail
2. size 변수

#### 큐의 연산

1. push(int x) : tail 에 원소 삽입
2. pop() : head 에 위치한 원소 삭제
3. front() : 제일 앞쪽의 원소 확인
4. back() : 제일 뒤쪽의 원소 확인

```

public class Main {
    // 배열 기반 큐를 위한 변수들
    static final int MX = 1000005;
    static int[] dat = new int[MX];
    static int head = 0;
    static int tail = 0;

    // 큐의 맨 뒤에 요소를 추가 (push)
    public static void push(int x) {
        dat[tail] = x;
        tail++;
    }

    // 큐의 맨 앞 요소를 제거 (pop)
    public static void pop() {
        head++;
    }

    // 큐의 맨 앞 요소를 반환 (front)
    public static int front() {
        return dat[head];
    }

    // 큐의 맨 뒤 요소를 반환 (back)
    public static int back() {
        return dat[tail - 1];
    }

    // 테스트 함수
    public static void test() {
        push(10);
        push(20);
        push(30);
        System.out.println(front()); // 10
        System.out.println(back()); // 30
        pop();
        pop();
        push(15);
    }
}

```

```
        push(25);
        System.out.println(front()); // 30
        System.out.println(back()); // 25
    }

    // 메인 함수
    public static void main(String[] args) {
        test();
    }
}
```

## 삽입 삭제가 빈번하다보면?

→ 공간의 낭비가 발생함.

그래서 등장한 개념이 원형 큐임!

## 원형 큐란?



앞과 뒤가 연결 되어있어서 head와 tail이 배열 안에서 움직이면서 공간의 낭비가 없도록 함.