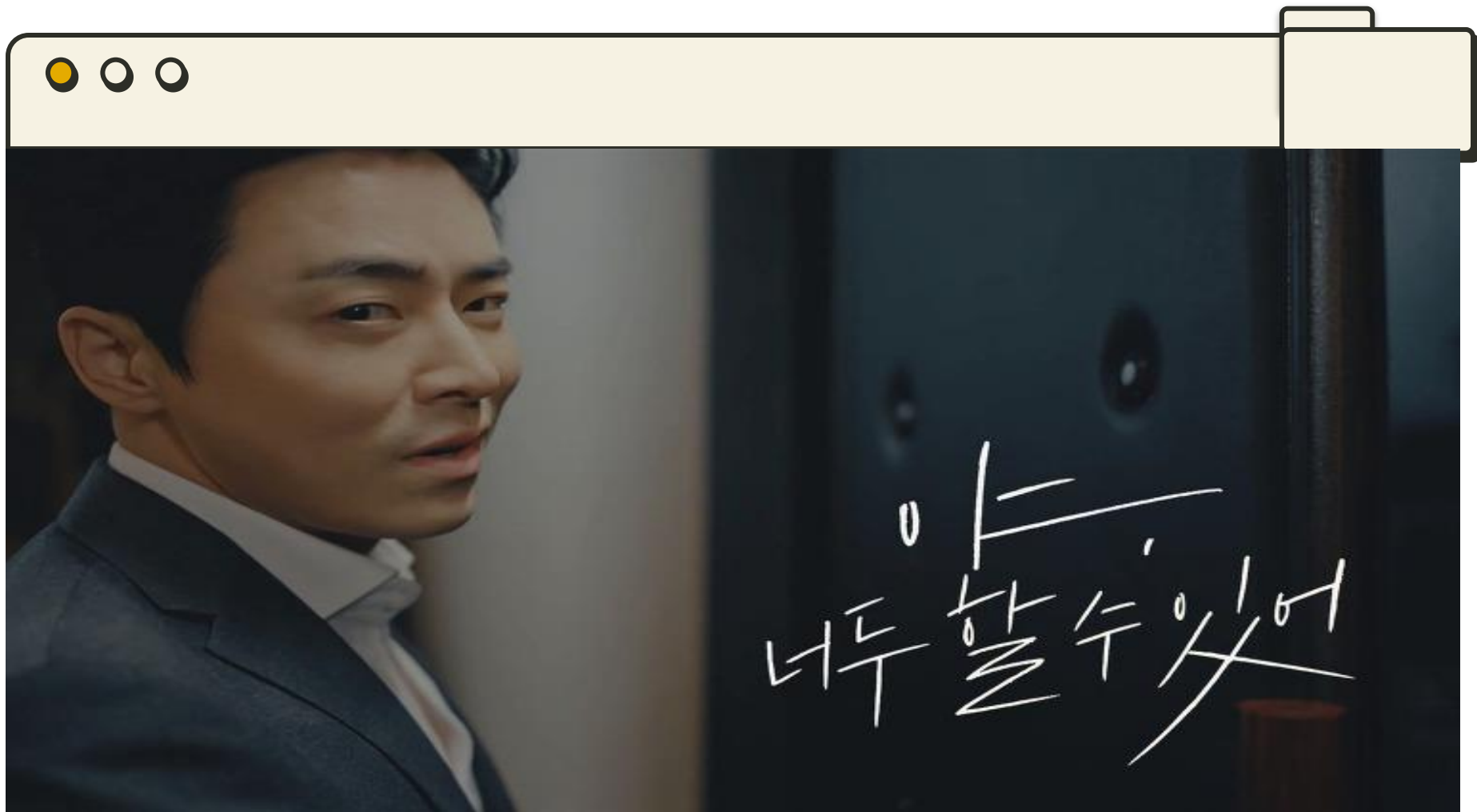


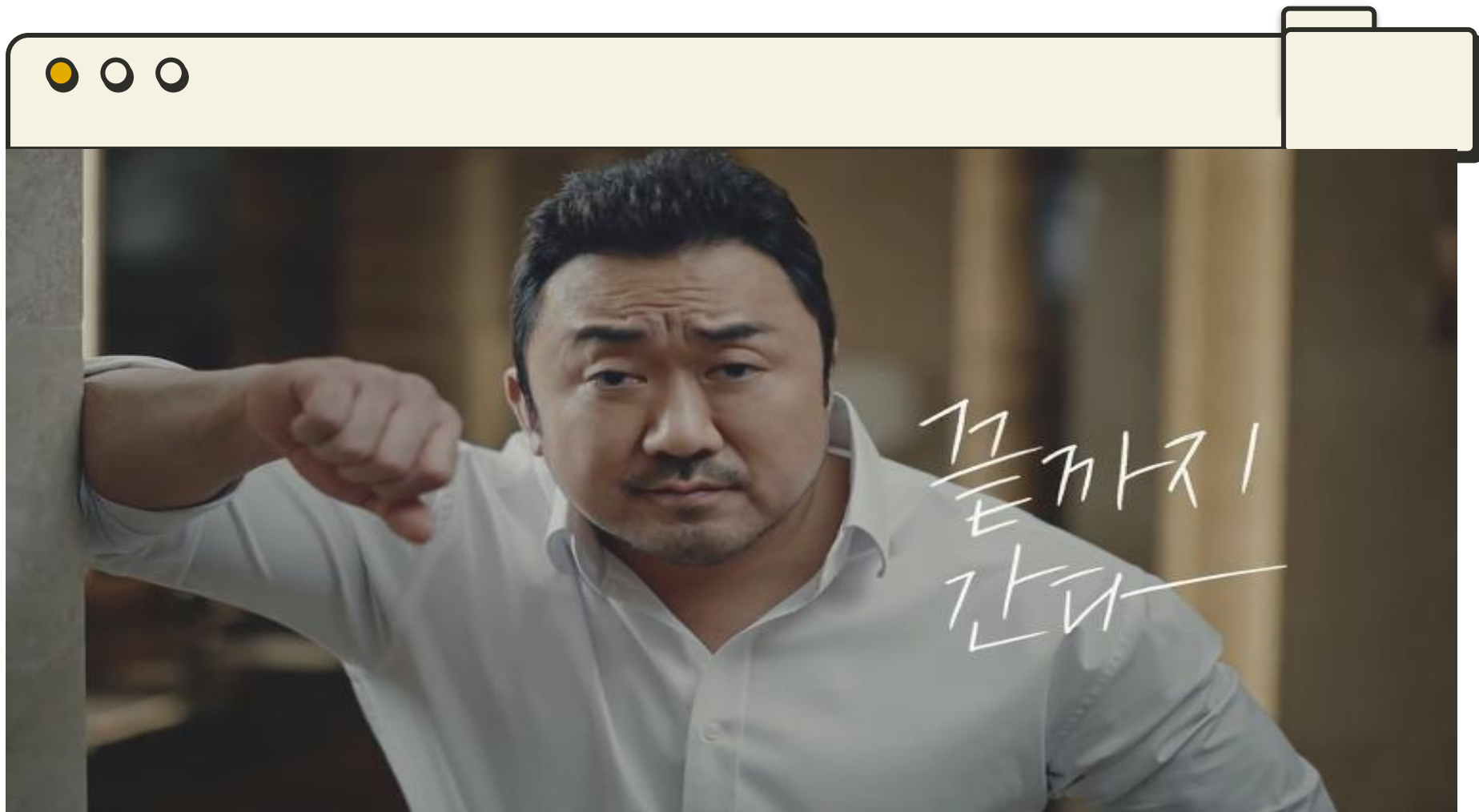


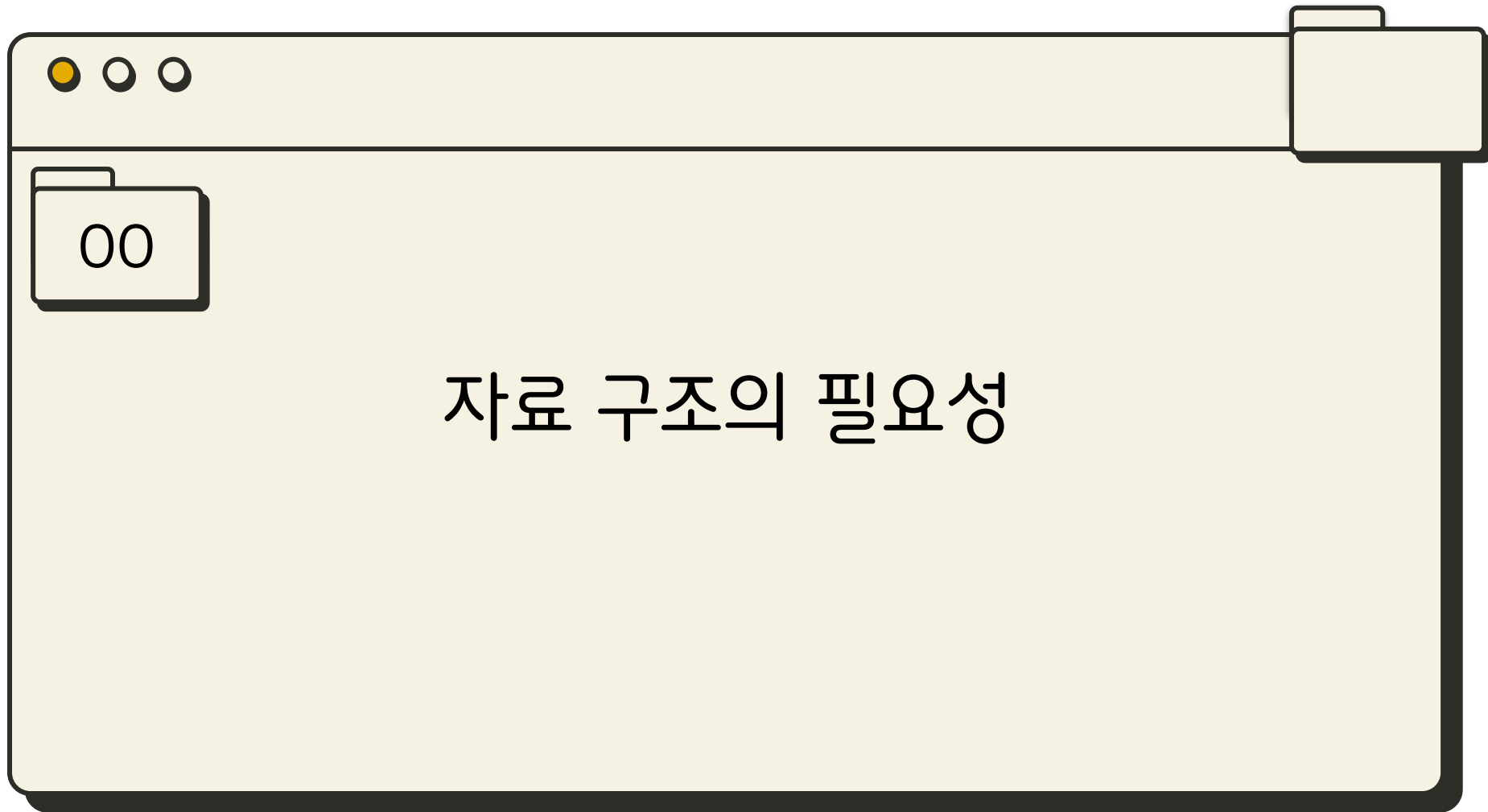
# 구현 & 시뮬레이션

Implementation & simulation

Here is where your presentation begins







# 자료 구조의 필요성

# 자료 구조의 필요성

## 컴퓨터와 자료구조

- ☑ 프로그램 : 데이터를 표현하고, 표현된 데이터를 처리하는 것
  - 데이터의 표현은 저장의 의미를 내포
- ☑ 자료를 어떻게 표현 여부에 따라 효과적인 알고리즘 개발이 좌우

## 자료구조

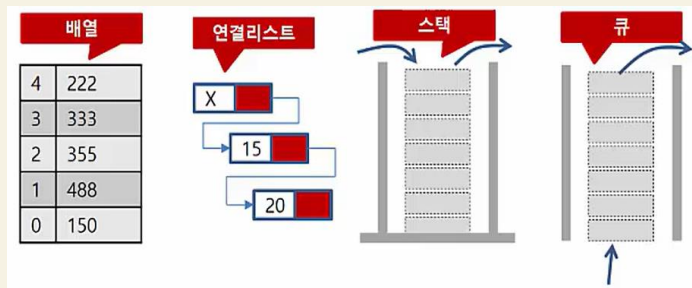
☒ 데이터를 저장하는 기억 공간의 형태

☒ 데이터를 참조, 추가, 삭제, 변경하는 방식

☒ 자료의 특성과 크기, 주요 사용법과 수행하는 연산의 종류, 구현에 필요한 기억 공간 크기에 따라 여러가지 종류의 자료구조 중 하나를 선택할 수 있음

## 선형 구조

- ★ 배열
- ★ 리스트
- ★ 스택
- ★ 큐



## 비선형 구조

- ★ 트리
- ★ 그래프



## 배열(Array)

- ☒ 번호(첨자(index))와 대응하는 데이터(원소)들로 이루어진 자료구조
- ☒ 일반적으로 같은 종류의 데이터들을 순차적으로 저장
- ☒ 첨자(index)는 배열의 시작점으로부터 각 데이터가 저장되어 있는 상대적 위치를 의미
- ☒ 임의 위치의 데이터에 대한 저장 및 참조 가능

0	1	2
92	91	87



## 리스트(List)

- ☑ 데이터를 순서대로 한 줄로 저장하는 선형 구조 자료
- ☑ head : 처음에 입력된 데이터의 위치
- ☑ Tail : 마지막에 입력된 데이터의 위치

## 배열 리스트(Array List)

- ☑ 연속된 저장 공간에 데이터를 저장하는 방식

d	a	t	a
head		tail	

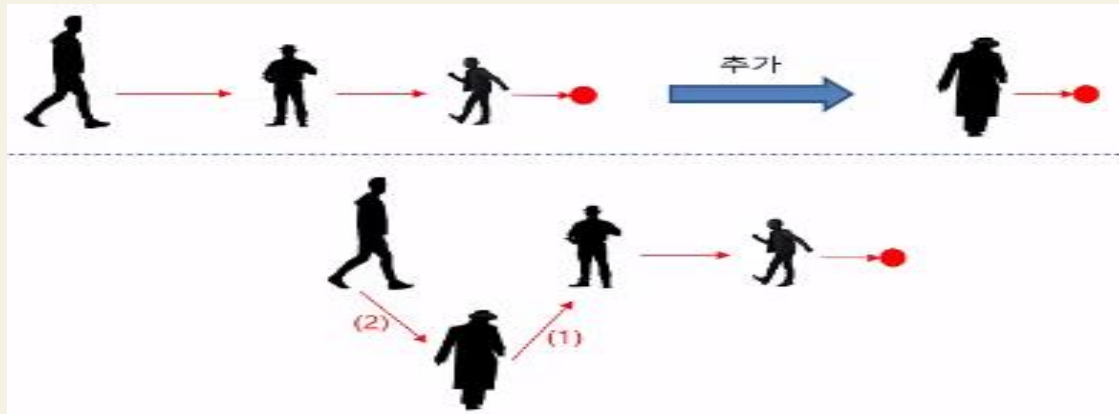
## 연결 리스트(Linked List)

- ☑ 서로 떨어져 있는 저장 공간에 데이터를 저장하되, 각 데이터와 함께 다음 데이터의 저장 공간 위치에 관한 정보를 함께 저장

	0	1	2	3	4	5	6	7	8	9	10	11
0	d	(1,2)					a	( )				
1			a	(1,9)						t	(0,6)	

## 리스트(List)

- ☑ 데이터의 저장 순서가 의미를 가짐
- ☑ 데이터를 중간 위치에 추가하기 위해서는 기존에 연결된 두 데이터의 연결을 끊고 새로 추가된 데이터와의 연결을 설정해야 함



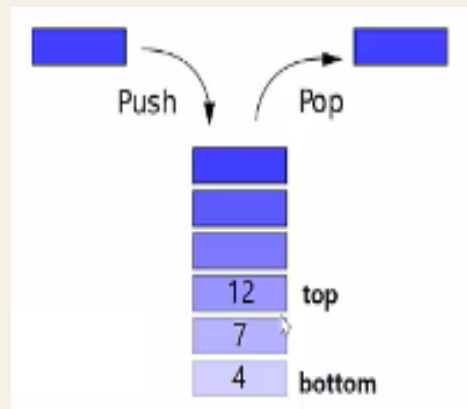
## 리스트(List)

☑ 구현에 따른 특징

연결리스트로 구현 할 경우, 복잡도는 증가하는 반면 추가 삭제를 위한  
오버헤드가 적음

## 스택(Stack)

- ☑ 제한적으로 접근할 수 있는 나열 구조
- ☑ 일련의 데이터 저장 공간의 한쪽 끝에서만 자료를 넣거나 뺄 수 있음
- ☑ 먼저 집어 넣은 데이터가 나중에 나오는 LIFO 구조
- ☑ 관련 용어
  - bottom : 가장 먼저 입력된 데이터가 저장된 위치
  - top : 가장 나중에 입력된 데이터가 저장된 위치
  - push : 스택에 새로운 데이터를 넣음
  - pop : 스택 top 위치의 데이터를 빼 냄

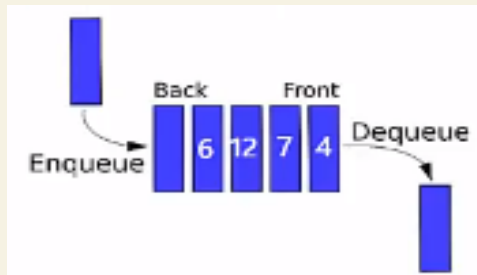


## 큐(Queue)

- ☑ 제한적으로 접근할 수 있는 나열 구조
- ☑ 일련의 데이터 저장 공간의 한쪽 끝에서만 자료를 넣고 반대편에서 자료를 뺄 수 있는 구조
- ☑ 먼저 집어 넣은 데이터가 나중에 나오는 FIFO 구조
- ☑ 관련 용어
  - Front : 데이터가 인출되는 위치
  - back : 데이터가 삽입되는 위치
  - enqueue : 큐에 자료를 넣는 연산, insert(삽입) 라고도 함
  - dequeue : 큐에서 자료를 꺼내는 연산, delete(인출) 라고도 함

데이터 입력 순서 :

4 -> 7 -> 12 -> 6



## 큐(Queue)

### 특수형태의 큐

- 데이터 저장 공간의 효율적인 사용을 위하여 데이터 접근 방법을 변형

### 원형 큐(circular queue)

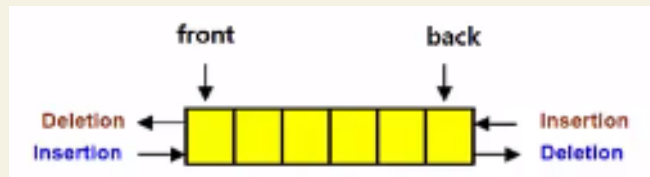
- 마지막 저장 공간에 자료가 삽입된 다음에는 첫번째 저장 공간부터 다시 큐를 채우는 형태

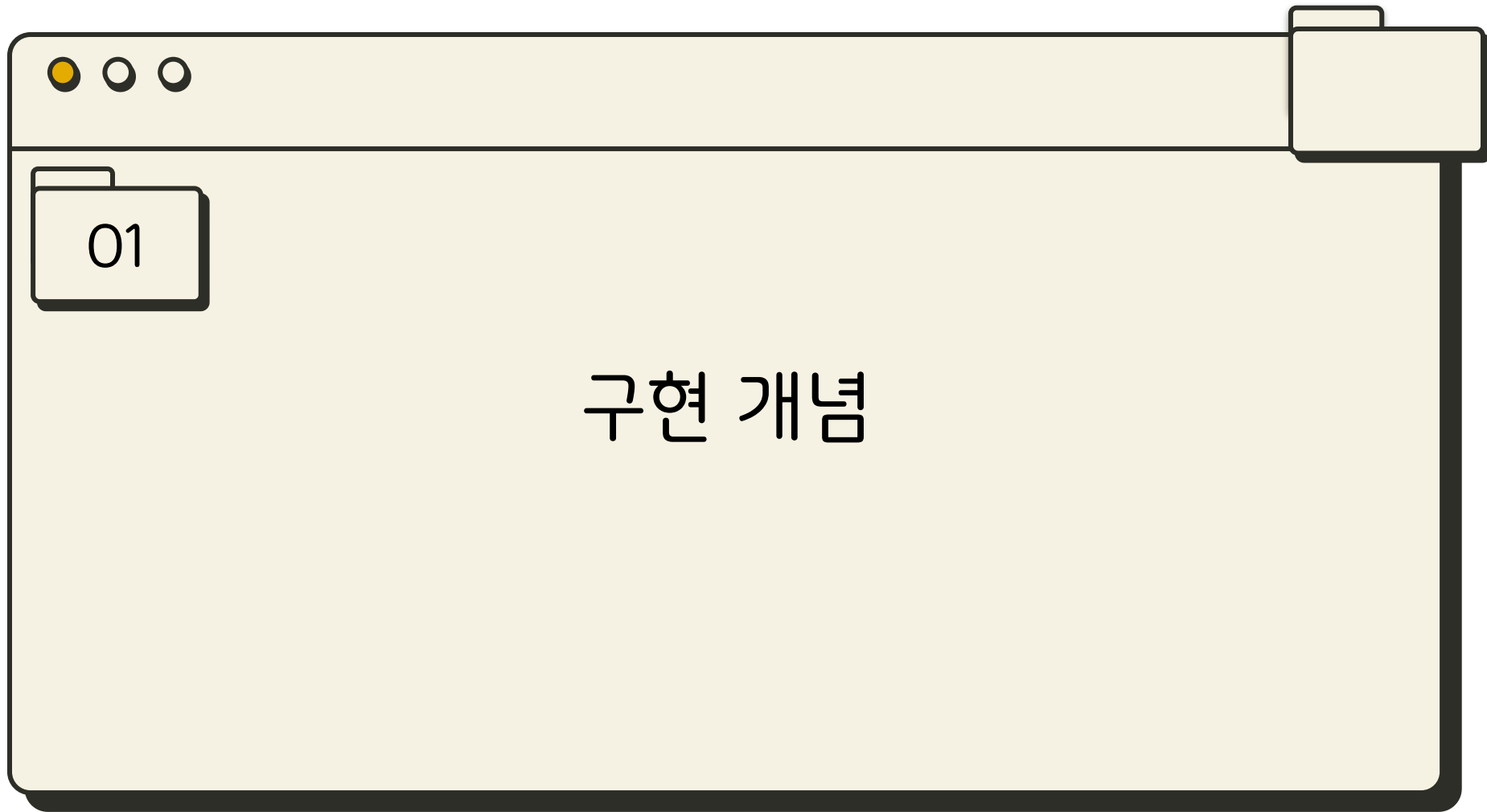
### 우선순위 큐

- 자료는 우선 순위와 함께 표현됨
- 자료는 순서대로 삽입되지만 인출은 우선순위에 따름

### 데크(deque)

- front와 back 모두에서 삽입과 인출이 가능

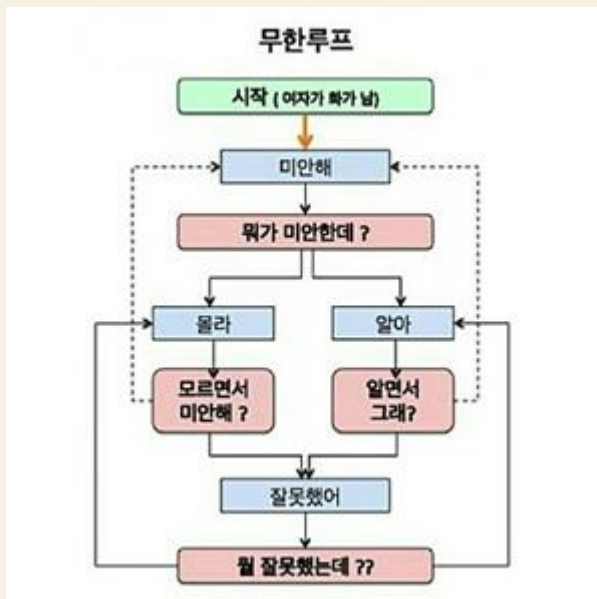




01

구현 개념

Q : 해당 프로세스를 구현하시오



:: 머릿속에 있는 알고리즘을 소스코드로  
바꾸는 과정

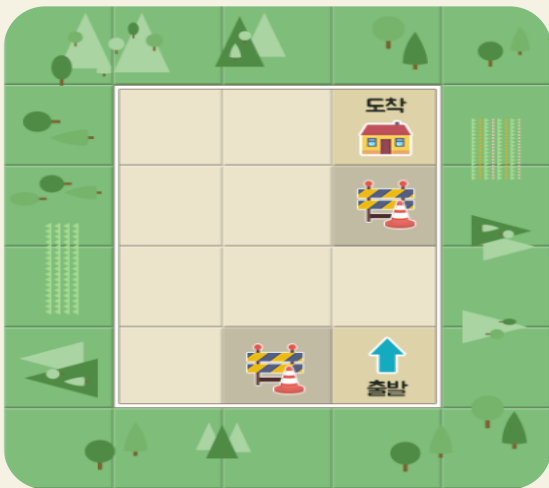




02

# 시뮬레이션 개념

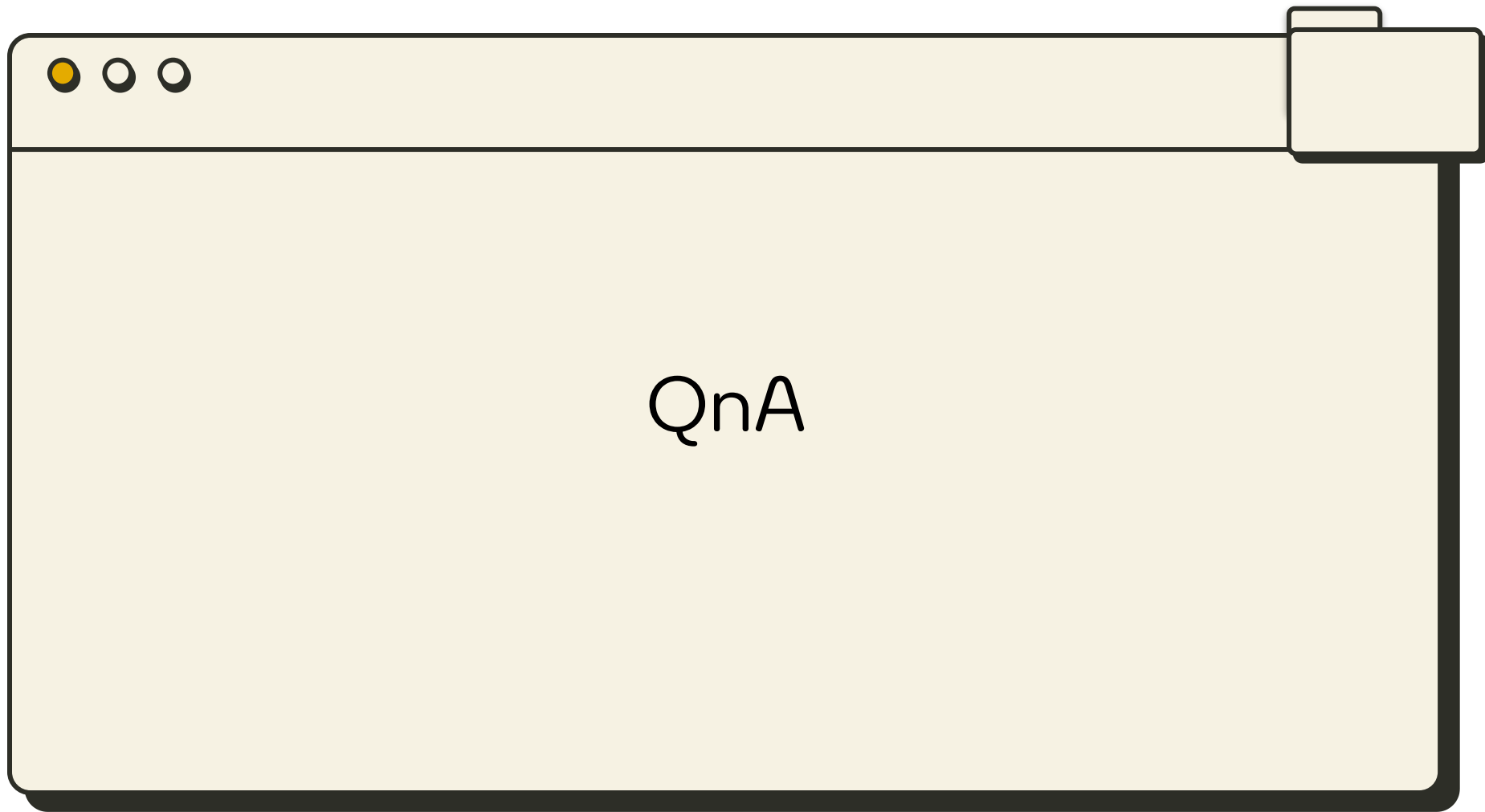
Q : 도착지까지 도달 하시오.



:: 각 조건에 맞는 상황을 구현하는 문제!

☒ 지도상에서 이동 및 탐험

☒ 배열 안에서 이동 및 탐험





# Reference

- 과학창의융합재단
- 충북대학교
- Harvard(하버드)