

# /\* 데이터 구조 및 알고리즘 \*/

신현규 강사, 화/목 20:00

3월14일 ~ 5월 4일



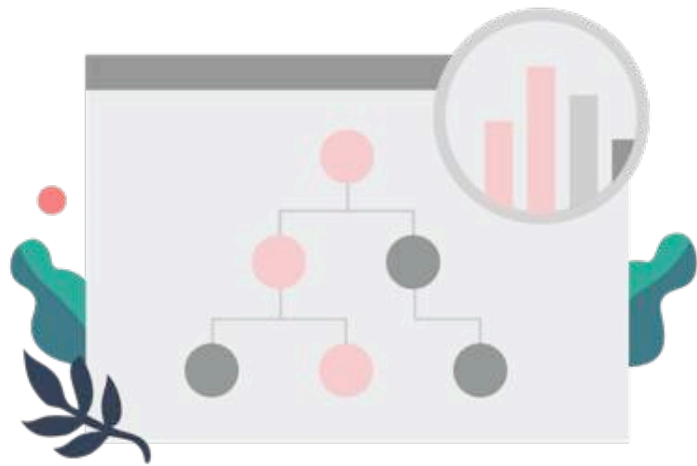
/\* elice \*/

# 목차

- 01 교육과정 소개
- 02 엘리스 사용법
- 03 배열, 연결리스트, 클래스

# 01 교육과정 소개

# /\* 데이터 구조 및 알고리즘 \*/



교육기간

3월 14일 ~ 5월 4일 (8주)

강사진

신현규 강사  
주민건 조교

라이브 강의

화/목 20:00 ~ 21:30

오피스 아워

월/수 20:00 ~ 21:30

# 수강 대상

프로그래밍을 깊이 있게 배우고 싶은 초보자

졸업 및 취업을 앞둔 전산학과 학생

코드의 성능을 끌어올리고 싶은 개발자

# 수강 대상

프로그래밍을 깊이 있게 배우고 싶은 초보자

→ 자료구조를 구현할 수 있는 프로그래밍 실력

졸업 및 취업을 앞둔 전산학과 학생

→ 자료구조를 필요에 따라 디자인 할 수 있는 능력

코드의 성능을 끌어올리고 싶은 개발자

→ 자료구조의 이용에 따른 성능 분석 능력

# 수강 목표

- 데이터 구조를 직접 Python 언어로 구현할 수 있다
- 주어진 문제를 효율적으로 해결하기 위한 자료구조를 디자인할 수 있다
- 자료구조의 이용에 따른 성능을 분석할 수 있다.

**본인의 수강 목표를 명확히 할 것**



# 강사진 소개



## 신현규 강사

POSTECH 컴퓨터공학과 최우등 졸업  
한국 정보올림피아드 경시대회 금상  
(전) 캘리포니아 오라클 연구소 소프트웨어 인턴  
(전) 대전창조경제혁신센터 데이터 사이언티스트  
과정 총괄강사

# 강사진 소개



## 주민건 조교

한양대학교 ERICA 컴퓨터공학과  
(전) IM4U 정보영재교육센터

# 주차별 커리큘럼

1주차	과정 소개, 배열, 연결리스트, 클래스
2주차	스택, 큐, 해싱
3주차	시간복잡도
4주차	트리, 트리순회, 재귀호출
5주차	힙
6주차	그래프 소개, DFS
7주차	그래프 심화, BFS
8주차	강의 요약, 알고리즘 과정 소개

## 02 엘리스 사용법

# 엘리스에서 코딩하기

모든 교육은 **엘리스**에서 진행 됩니다.

The screenshot displays the Elice Academy web application. The browser address bar shows the URL `academy.elice.io/courses/18/lectures/24/materials/3`. The page title is "실습문제 2 - Default". The sidebar on the left contains the following text:

이상한 코딩의 나라에 어서 오세요!

하트 여왕이 힌토끼에게 사형수 목록을 출력하는 프로그램을 만들어 달라고 했습니다, 방금 배운 `print` 를 이용해서 사형수를 출력하는 프로그램을 만들어 볼까요?

**설명**

출력할 때에는

```
print("원하는 출력 내용")
```

이라고 쓰면 됩니다.

1. 다음 사형수 목록을 하나씩 출력해 보세요!

The central code editor shows a file named `main.py` with the following code:

```
print("하트 여왕의 사형수 목록: ")
```

The right sidebar features a live chat window with a purple cat avatar. Below the chat window, there are buttons for "실행" (Run), "재출" (Re-run), and "입력 데이터" (Input Data). The status bar at the bottom indicates "아직 출력 결과가 없습니다." (No output results yet).

# 화면 레이아웃 바꾸기

‘극장 모드’와 ‘전체 화면’을 사용하면 강의 영상을 더욱 크게 보실 수 있습니다.

The screenshot shows the elice.io web interface. The browser address bar displays `academy.elice.io/courses/18/lectures/24/materials/3`. The page title is "실습문제 2 - Default". The sidebar on the left contains the text "이상한 코딩의 나라에 어서 오세요!" and a description of a programming task. The main area shows a code editor with a Python snippet: `print("하트 여왕의 사형수 목록: ")`. The video player on the right shows a purple cat character. Two circular callouts highlight the "극장 모드" (Theater Mode) and "전체 화면" (Full Screen) icons in the video player controls.

# 화면 레이아웃 바꾸기

‘와이드 모드’로 강의를 시청하며  
동시에 코드 작성이 가능합니다.

The screenshot shows the Elice Academy web interface. On the left is a lesson description in Korean, titled '이상한 코딩의 나라에 어서 오세요!' (Welcome to the Strange Country of Coding). The main area features a code editor with a Python file named 'main.py' containing the line `print("하트 여왕의 사형수 목록: ")`. To the right of the code editor is a video player showing a purple cat character. A circular callout with a white border highlights a button labeled '와이드 모드' (Wide Mode) located over the video player. The interface also includes a top navigation bar with '강의 목록' (Lesson List) and '실습문제 2 - Default', and a bottom bar with buttons for '실행' (Run), '재출' (Re-run), and '입력 데이터' (Input Data).

# 도움요청

실습문제를 푸는 도중 도움이 필요하시면  
‘도움 요청’을 해보세요.

The screenshot shows the Elice IDE interface. On the left, a sidebar contains the exercise description in Korean: '이상한 코딩의 나라에 어서 오세요!' (Welcome to the strange coding country!), a problem statement about a queen's subject list, a '설명' (Description) section, and a code example: `print("원하는 출력 내용")`. The main area displays a Python file named `main.py` with the code: `print("하트 여왕의 사형수 목록: ")`. A toolbar above the code editor includes buttons for '도와주세요!' (Help), a checkmark, a clock, and a '도움 요청' (Request Help) button. The bottom status bar shows '실행' (Run), '제출' (Submit), '입력 데이터' (Input Data), and '최고 점수 :-' (Best Score :-). The output area at the bottom states '아직 출력 결과가 없습니다.' (No output results yet).



# 도움요청

도와주세요 버튼을 누르세요.

The screenshot shows the elice web application interface. The browser tab is titled 'elice' and the address bar shows '0.0.0.0:8080/courses/18/lectures/24/materials/5'. The page header includes a navigation menu with '강의 목록' (Lecture List) and '문제4 \_ Default' (Problem 4 \_ Default). The user is logged in as 'elice student'.

The main content area displays a coding problem titled '이상한 코딩의 나라에 어서 오세요!' (Welcome to the strange coding country!). The problem description asks the user to write a program that prints the list of data types for the Queen of Hearts. The code editor shows a single line of Python code: `print("하트 여왕의 사형수 목록: ")`. A large, semi-transparent circular overlay with a white border is positioned over the code editor, containing the text '도와주세요!' (Ask for help!).

The left sidebar contains the problem description and a '설명' (Explanation) section. The '설명' section states: '출력할 때에는 print("원하는 출력 내용")' (When outputting, use print("desired output content")).

The bottom of the interface features a toolbar with buttons for '실행' (Run), '제출' (Submit), and '입력 데이터' (Input Data). The '제출' button is highlighted in green. To the right of the toolbar, there is a section for '최고 점수 :-' (Highest score :-) and a '도와요청' (Request help) button.

# 도움요청

바로 **도움 요청 창**으로 이동합니다.  
어떤 도움이 필요하신지 작성해주세요.

The screenshot shows the Elice IDE interface. On the left, there is a sidebar with a menu icon and the text '강의 목록'. The main area is divided into two panels. The left panel contains text about a programming exercise: '이상한 코딩의 나라에 어서 오세요!', a description of the task (printing a list of queen's suitors), and a code snippet: `print("원하는 출력 내용")`. The right panel shows a code editor with a file named 'main.py' containing the code: `print("하트 여왕의 사형수 목록: ")`. Below the code editor, there are buttons for '실행' (Run), '제출' (Submit), and '입력 데이터' (Input Data). A '도움요청' (Request Help) button is visible in the top right corner of the IDE. A large, semi-transparent circular overlay is positioned over the bottom right of the IDE, containing a blue button with the text '도움 요청' (Request Help).

# 도움요청

바로 **도움 요청 창**으로 이동합니다.  
어떤 도움이 필요하신지 작성해주세요.

The screenshot shows the elice.io web interface. On the left, there's a sidebar with a problem description in Korean: "이상한 코딩의 나라에 어서 오세요!" (Welcome to the strange coding country!). The main area displays a code editor with a Python file named "main.py" containing the line `print("하트 여왕의 사형수 목록: ")`. Below the code editor are buttons for "실행" (Run), "제출" (Submit), and "입력 데이터" (Input Data). On the right, there's a "도움 요청" (Request Help) button. A circular overlay on the right side of the screen shows a chat window with a pink rabbit avatar and the text: "elice student 7분 도와주세요 ㅠㅠ" (elice student 7 minutes help please ㅠㅠ) and "메시지를 입력하" (Enter message).

# 도움요청

도움 요청 게시글 아래 댓글을 통해  
**조교와 대화**를 나누실 수 있습니다.

The screenshot displays the Elice Academy web application. On the left, a sidebar contains a lesson titled "이상한 코딩의 나라에 어서 오세요!" (Welcome to the Strange Country of Coding). The main content area shows a Python code editor with a file named "main.py" containing the line `print("하트 여왕의 사형수 목록: ")`. Below the code editor are buttons for "실행" (Run), "제출" (Submit), and "입력 데이터" (Input Data). On the right, a chat window is open, showing a message from "elice student" asking for help and a response from "Elice Instructor" providing instructions. A circular callout highlights the chat window.

이상한 코딩의 나라에 어서 오세요!

하트 여왕이 힌토끼에게 사형수 목록을 출력하는 프로그램을 만들어 달라고 했습니다, 방금 배운 `print` 를 이용해서 사형수를 출력하는 프로그램을 만들어 볼까요?

**설명**

출력할 때에는

```
print("원하는 출력 내용")
```

이라고 쓰면 됩니다.

main.py × 도와주세요! ✓ ⌚ ⋮

```
1 print("하트 여왕의 사형수 목록: ")
```

실행 제출 입력 데이터 최고 점수 :-

아직 출력 결과가 없습니다.

elice student 8분 전 도와주세요 ㅠㅠ

Elice Instructor 몇 초 이렇게 해보세요! :)

메시지를 입력하

# 헬프센터

요청 게시글은 **헬프 센터**에서  
확인하실 수 있습니다.

The screenshot shows a web browser window with the URL `academy.elice.io/courses/18/helpcenter/none/none/status/all/feed/66/`. The page title is `/* elice */` and the navigation bar includes `나의 수강과목` and `과목 목록`. The main header is `기초 파이썬 프로그래밍` with sub-navigation for `과목 정보`, `강의`, `게시판`, and `헬프 센터`. The `헬프 센터` section is highlighted in a circular callout, showing a purple header with `헬프 센터` and a user profile for `Elice Instructor`. The main content area shows a chat interface with a user `elice student` and an `Elice Instructor`. The chat history includes a message from the student: `print("하트 여왕의 사형수 목록: ")` and a response from the instructor: `이렇게 해보세요! :)`. The chat is dated `17년 3월 13일` and `18시 4분`. The bottom of the page shows a text input field with the placeholder `메시지를 입력하세요.` and a page number `21`.

# 헬프센터

해결되었다면 조교 평가와 함께  
도움 **완료하기** 버튼을 클릭해주세요.

The screenshot displays the Elice Help Center interface. At the top, the browser address bar shows the URL: `academy.elice.io/courses/18/helpcenter/none/none/status/all/feed/66/`. The page header includes the text `/* elice */` and navigation links: `나의 수강과목` and `과목 목록`. The main header is titled `기초 파이썬 프로그래밍` and contains tabs for `과목 정보`, `강의`, `게시판`, and `헬프 센터`. The chat interface shows a conversation between `elice student` and `Elice Instructor`. The chat history includes a message from the instructor: `Elice Instructor: 이렇게 해보세요! :)` and a code snippet: `print("하트 여왕의 사형수 목록: ")`. A feedback modal is overlaid on the chat, asking `TA가 얼마나 도움이 되었나요?` with a 5-star rating system. The modal includes buttons for `평가하고 완료하기` and `취소`. The page number `22` is visible at the bottom.



# 03 배열, 연결리스트, 클래스

<데이터 구조>란?





# 데이터 구조 (자료구조)

= 데이터를 저장하는 구조

왜 데이터구조가 중요한가?

컴퓨터 = 자료 저장 + 연산

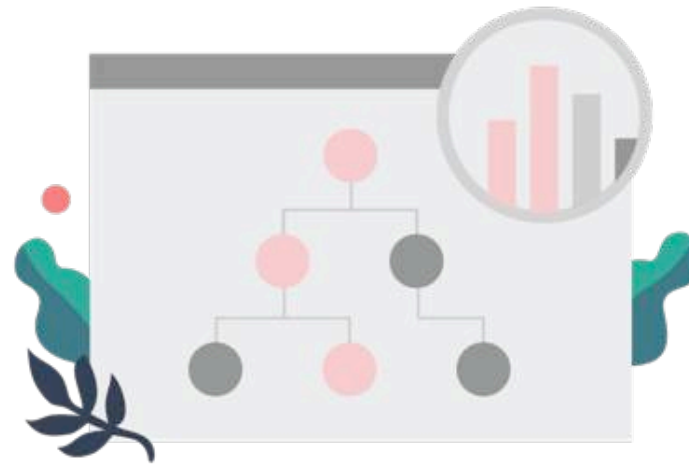
# 컴퓨터 공학의 기본 커리큘럼



## 1. 프로그래밍 언어

C / C++  
Python  
Matlab

# 컴퓨터 공학의 기본 커리큘럼



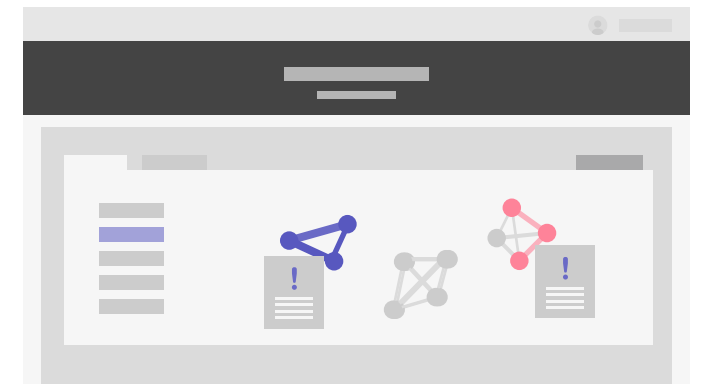
## 1. 프로그래밍 언어

C / C++  
Python  
Matlab

## 2. 자료구조

Stack  
Queue  
Tree

# 컴퓨터 공학의 기본 커리큘럼



## 1. 프로그래밍 언어

C / C++  
Python  
Matlab

## 2. 자료구조

Stack  
Queue  
Tree

## 3. 알고리즘

Brute-Force  
Divide & Conquer  
Dynamic Programming

# 샴푸 통 디자인

# 샴푸 통 디자인





# 샴푸 통 디자인



# 샴푸 통 디자인



# 샴푸 통 디자인



# 샴푸 통 디자인



+ 디자인이 간단

+ 쓰기 좀 더 편함

+ 쓰기 간편함

+ 우주에서 쓸수있음

- 쓰기가 다소 불편

- 하지만 여전히 불편함

- 많은 양을 얻기 힘들

- 중력이 있으면 굳이..

이용 목적에 따라 그 효율성이 다름

# 변수 (Variable)

= 가장 기본적인 자료구조

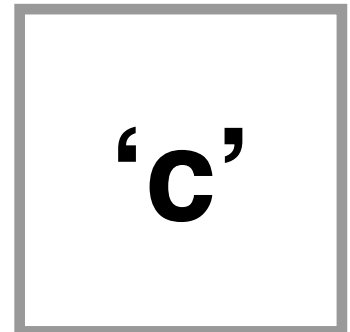
```
x = 1  
y = 'c'
```

```
print x  
print y
```

x



y



# 리스트 (List)

= 변수의 나열

	0	1	2	3	4	5	6	7
myList	3	4	2	5	1	2	6	3

# 리스트 (List)

= 변수의 나열

	0	1	2	3	4	5	6	7
myList	3	4	2	5	1	2	6	3



장점



# 리스트 (List)

= 변수의 나열

	0	1	2	3	4	5	6	7
myList	3	4	2	5	1	2	6	3



**장점**

i 번째 원소를 바로 알 수 있다 (myList[i])

# 리스트 (List)

= 변수의 나열

	0	1	2	3	4	5	6	7
myList	3	4	2	5	1	2	6	3



단점

# 리스트 (List)

= 변수의 나열

	0	1	2	3	4	5	6	7
myList	3	4	2	5	1	2	6	3



**단점**

원소의 추가/ 삭제가 까다롭다

# 리스트 (List)

= 변수의 나열

	0	1	2	3	4	5	6	7
myList	3	4	2	5	1	2	6	3

4번째에 숫자 10을 추가하자



**단점**

원소의 추가/ 삭제가 까다롭다

# 리스트 (List)

= 변수의 나열

	0	1	2	3	4	5	6	7	8
myList	3	4	2	5	1	2	6	3	

4번째에 숫자 10을 추가하자

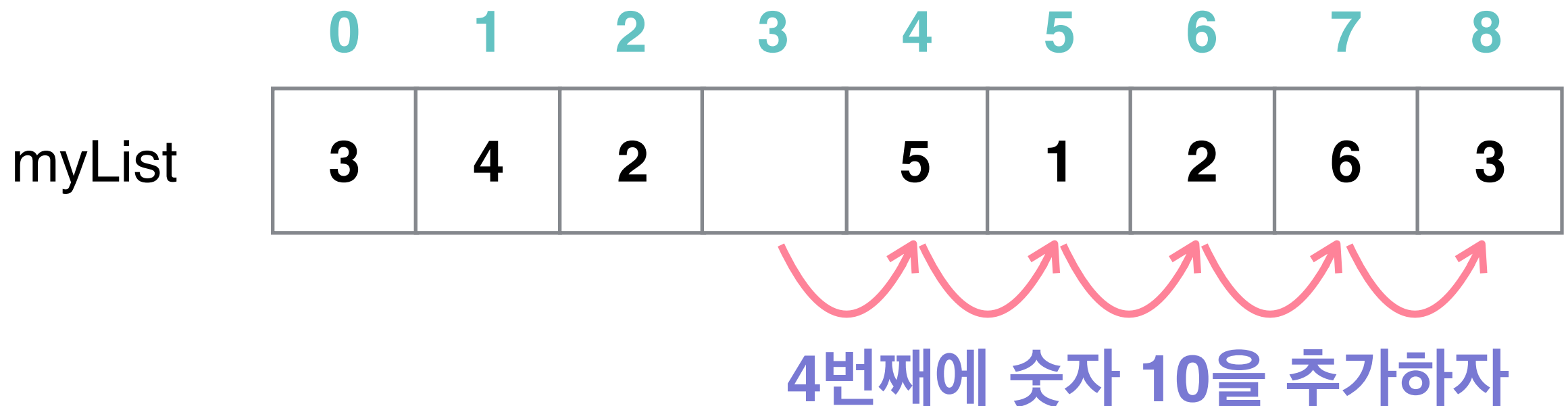


**단점**

원소의 추가/ 삭제가 까다롭다

# 리스트 (List)

= 변수의 나열



**단점**

원소의 추가/ 삭제가 까다롭다

# 리스트 (List)

= 변수의 나열

	0	1	2	3	4	5	6	7	8
myList	3	4	2	10	5	1	2	6	3

4번째에 숫자 10을 추가하자



**단점**

원소의 추가/ 삭제가 까다롭다

# 리스트 (List)

= 변수의 나열

	0	1	2	3	4	5	6	7	8
myList	3	4	2	10	5	1	2	6	3

5번째 숫자를 제거하자



**단점**

원소의 추가/ 삭제가 까다롭다



# 리스트 (List)

= 변수의 나열

	0	1	2	3	4	5	6	7	8
myList	3	4	2	10		1	2	6	3

5번째 숫자를 제거하자

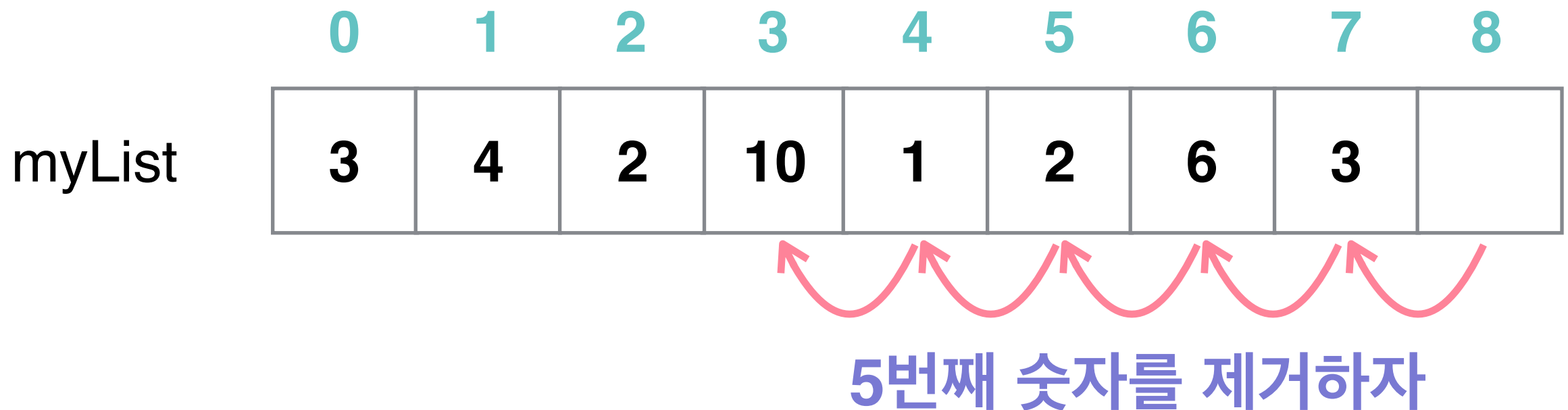


**단점**

원소의 추가/ 삭제가 까다롭다

# 리스트 (List)

= 변수의 나열



**단점**

원소의 추가/ 삭제가 까다롭다

# 리스트 (List)

= 변수의 나열

	0	1	2	3	4	5	6	7
myList	3	4	2	10	1	2	6	3

5번째 숫자를 제거하자



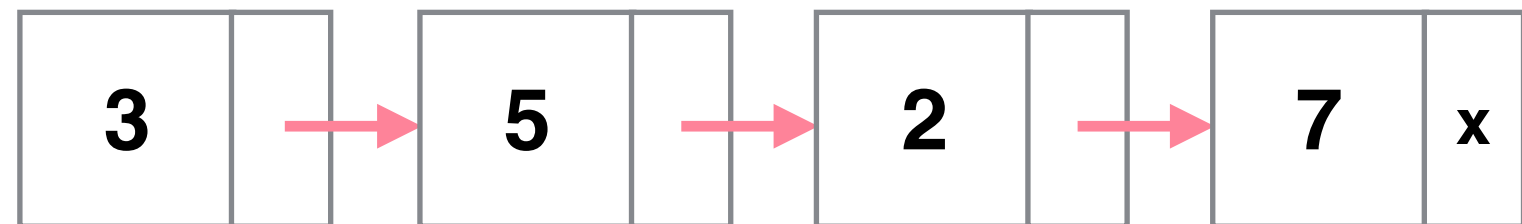
**단점**

원소의 추가/ 삭제가 까다롭다

# 링크드 리스트 (Linked List)

= 여러 개의 변수를 저장하는 다른 방법

myLinkedList

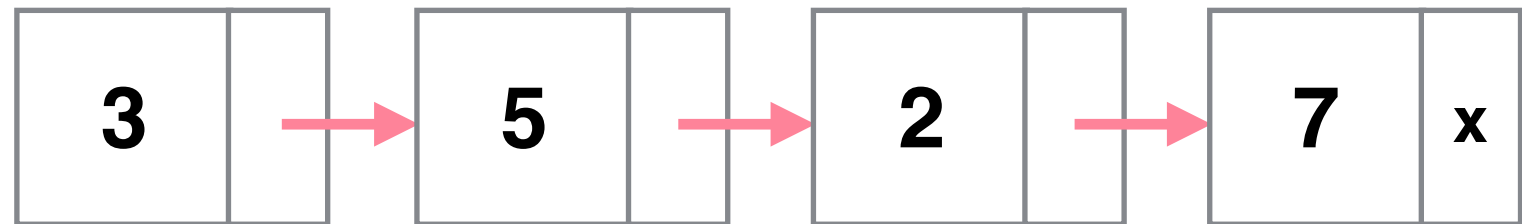


# 링크드 리스트 (Linked List)

= 여러 개의 변수를 저장하는 다른 방법

## 변수추가

myLinkedList



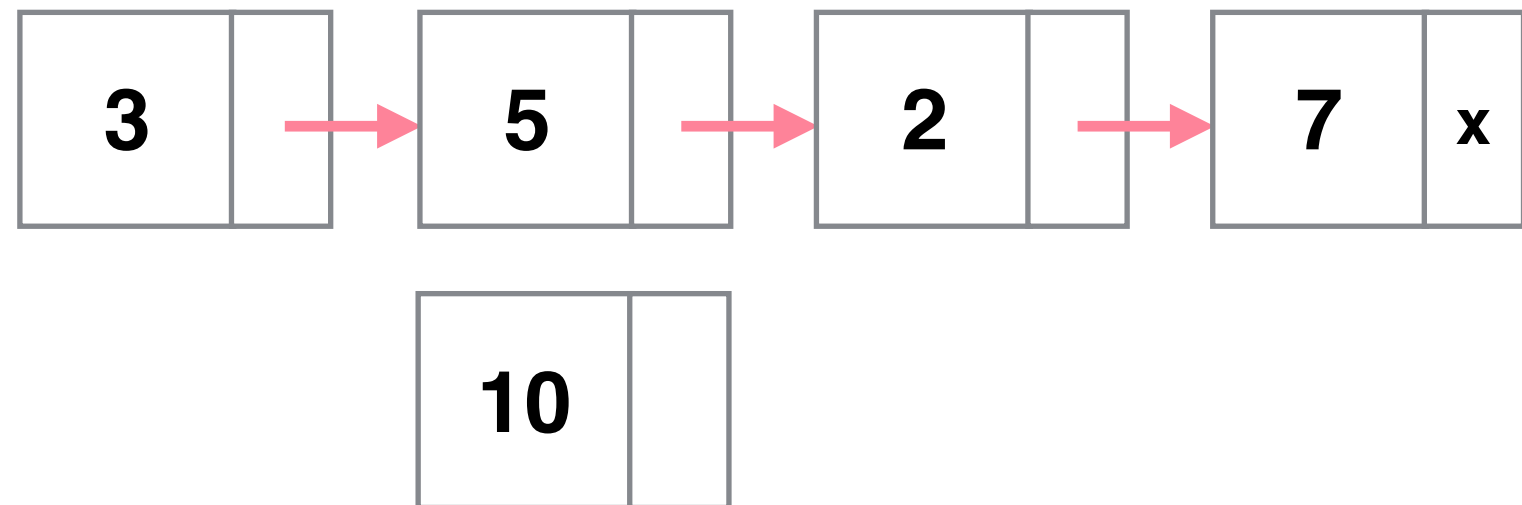
3번째에 숫자 10을 추가하자

# 링크드 리스트 (Linked List)

= 여러 개의 변수를 저장하는 다른 방법

## 변수추가

myLinkedList



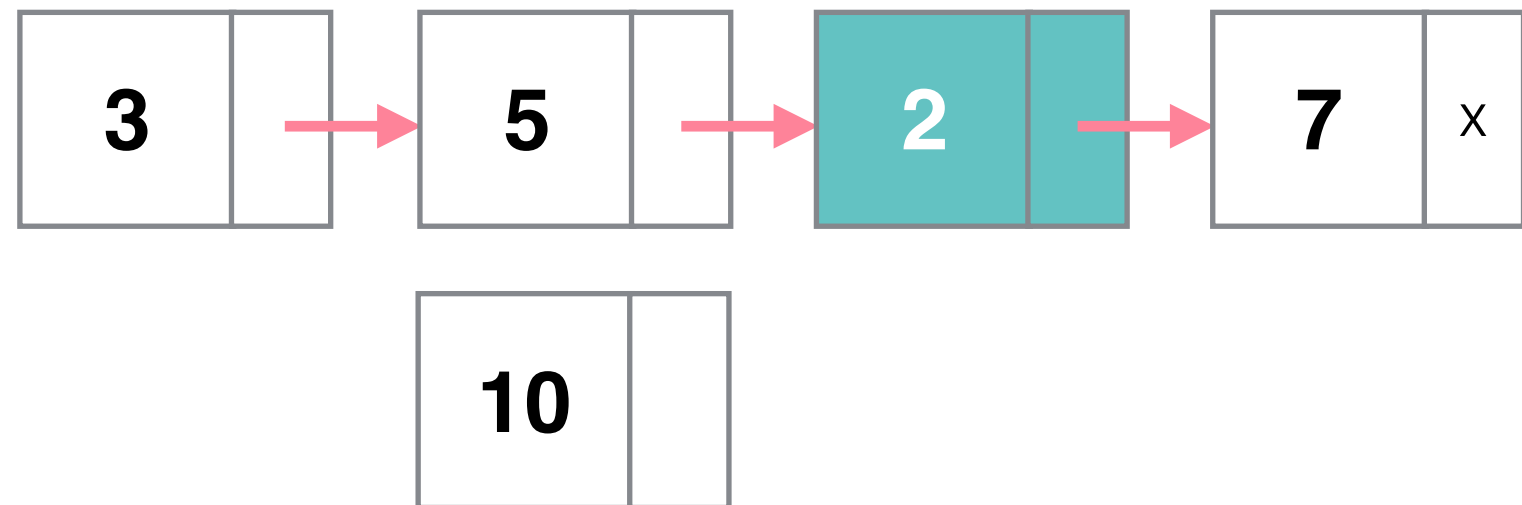
3번째에 숫자 10을 추가하자

# 링크드 리스트 (Linked List)

= 여러 개의 변수를 저장하는 다른 방법

## 변수추가

myLinkedList



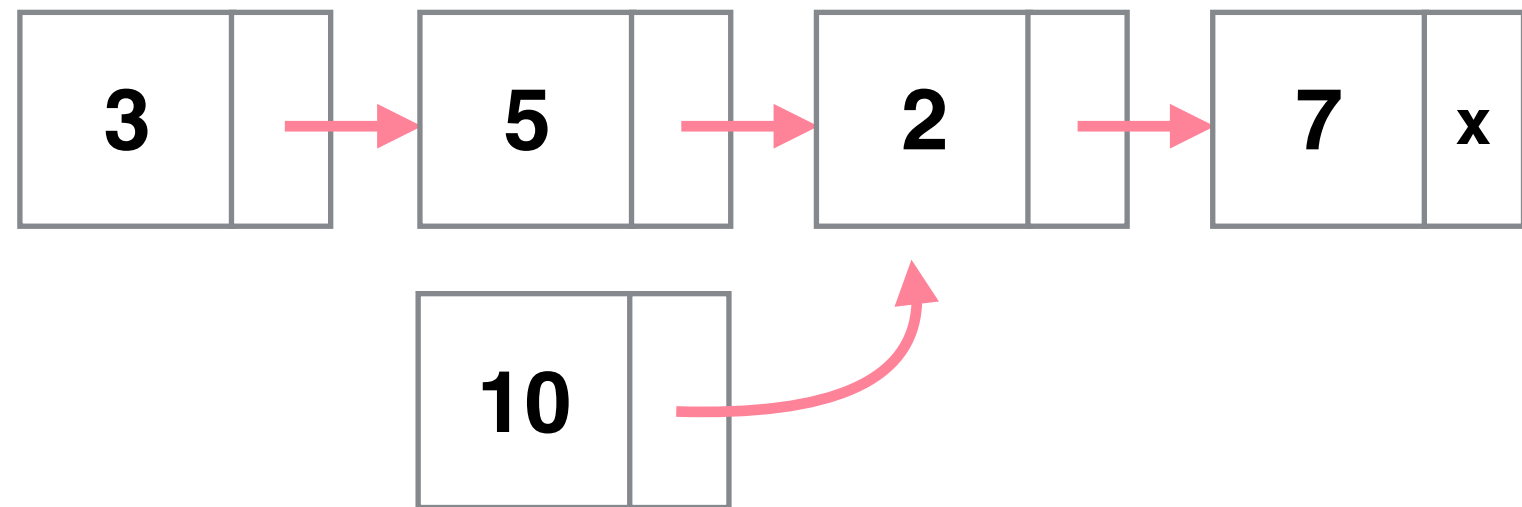
3번째에 숫자 10을 추가하자

# 링크드 리스트 (Linked List)

= 여러 개의 변수를 저장하는 다른 방법

## 변수추가

myLinkedList



3번째에 숫자 10을 추가하자

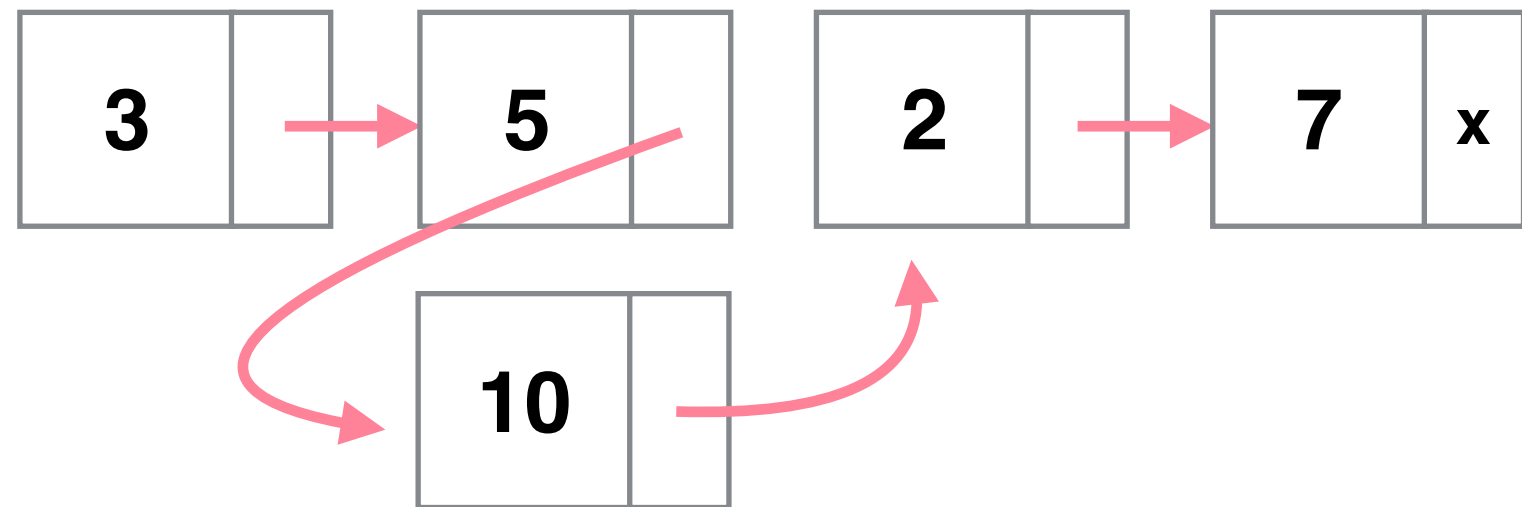


# 링크드 리스트 (Linked List)

= 여러 개의 변수를 저장하는 다른 방법

## 변수추가

myLinkedList



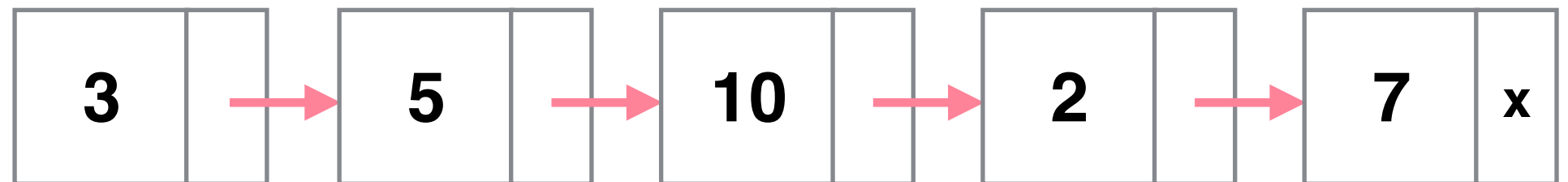
3번째에 숫자 10을 추가하자

# 링크드 리스트 (Linked List)

= 여러 개의 변수를 저장하는 다른 방법

변수추가

myLinkedList

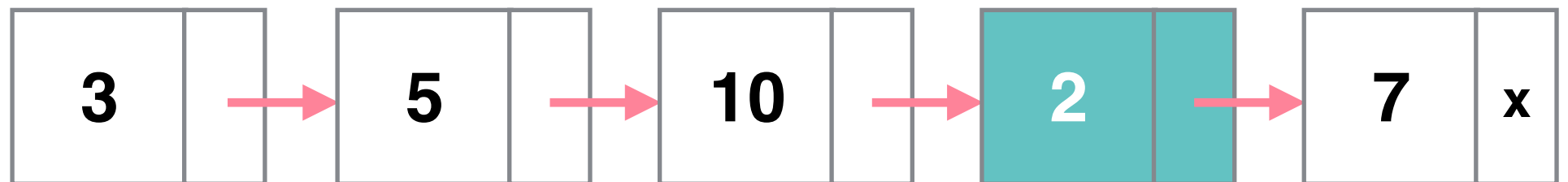


# 링크드 리스트 (Linked List)

= 여러 개의 변수를 저장하는 다른 방법

## 변수제거

myLinkedList



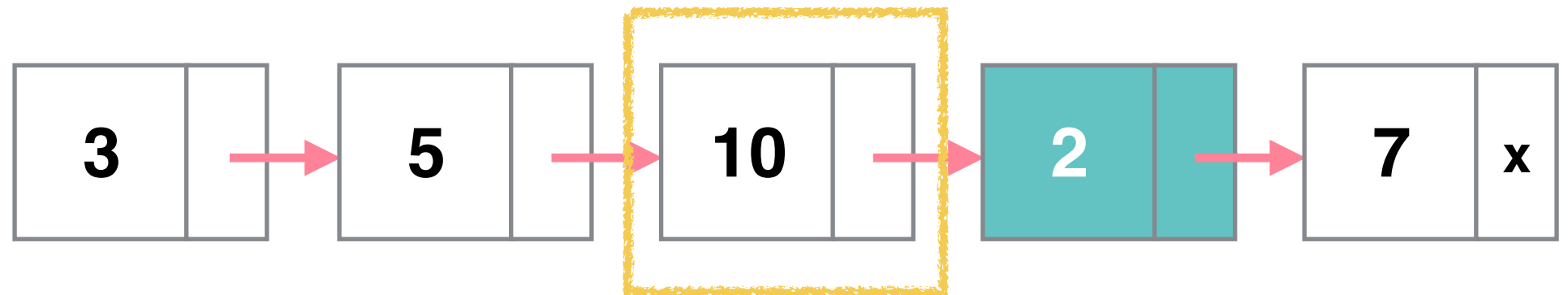
4번째에 숫자 제거하자

# 링크드 리스트 (Linked List)

= 여러 개의 변수를 저장하는 다른 방법

## 변수제거

myLinkedList



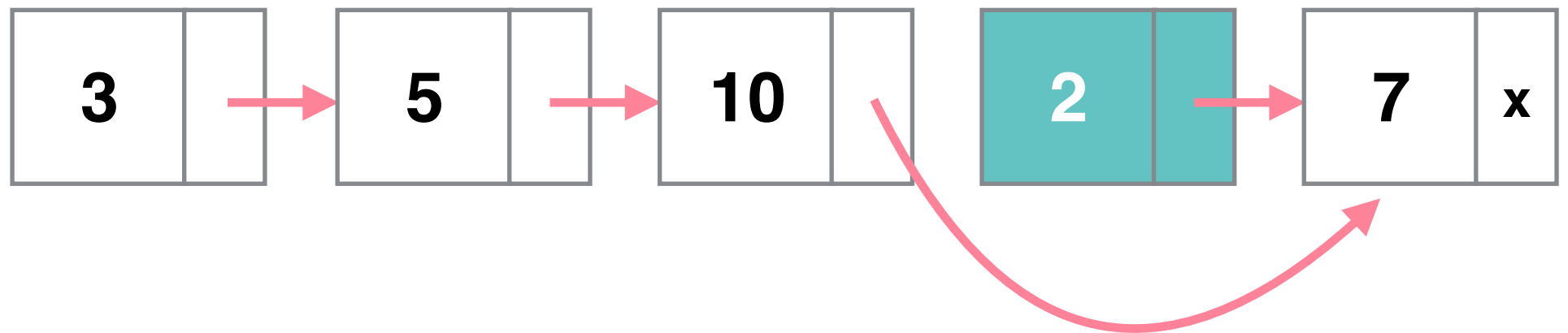
4번째에 숫자 제거하자

# 링크드 리스트 (Linked List)

= 여러 개의 변수를 저장하는 다른 방법

## 변수제거

myLinkedList



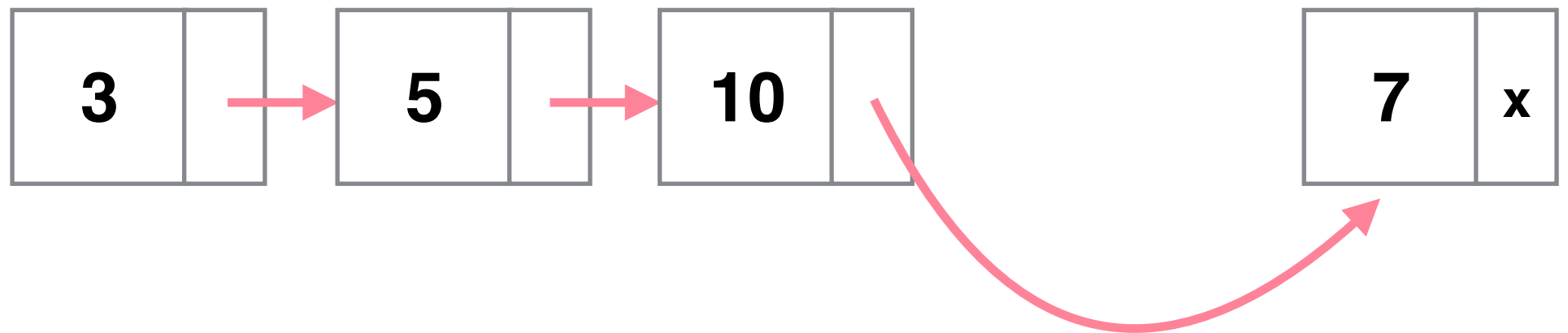
4번째에 숫자 제거하자

# 링크드 리스트 (Linked List)

= 여러 개의 변수를 저장하는 다른 방법

## 변수제거

myLinkedList



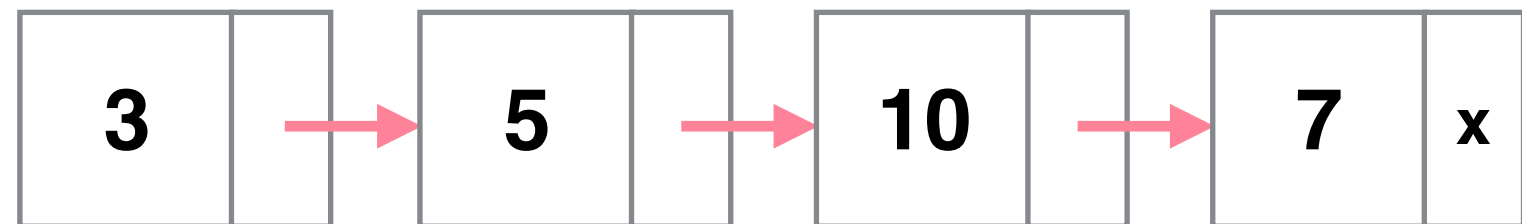
4번째에 숫자 제거하자

# 링크드 리스트 (Linked List)

= 여러 개의 변수를 저장하는 다른 방법

변수제거

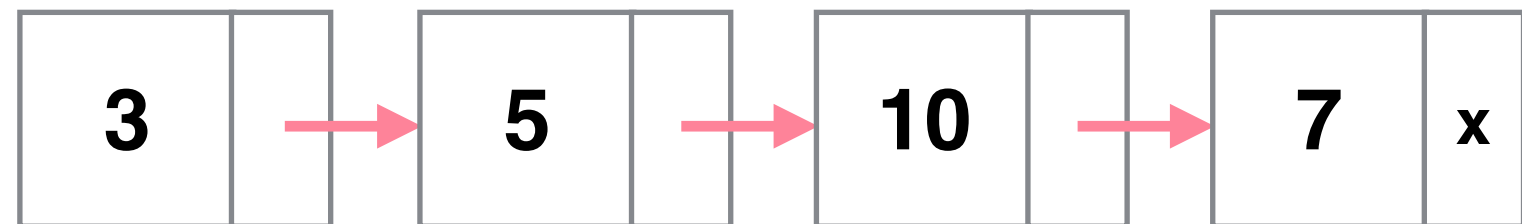
myLinkedList



# 링크드 리스트 (Linked List)

예제) 2번째에 숫자 11을 추가하세요.

myLinkedList

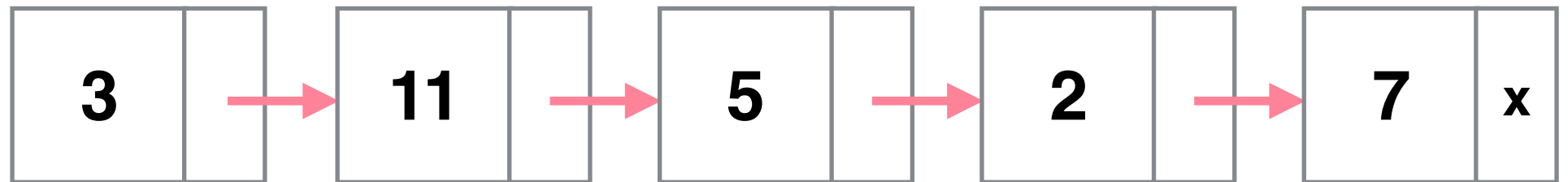




# 링크드 리스트 (Linked List)

예제) 2번째에 숫자 11을 추가하세요.

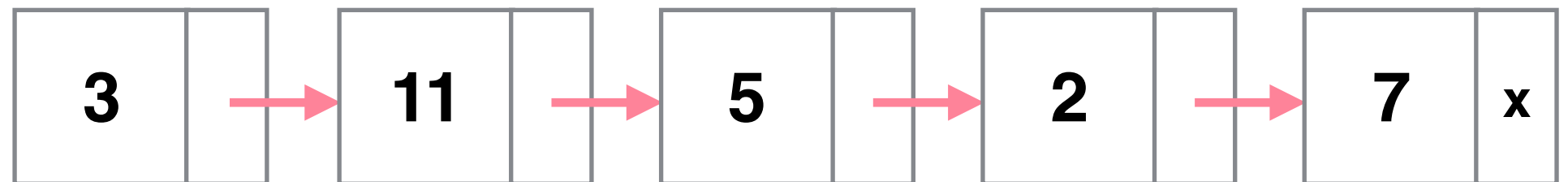
myLinkedList



# 링크드 리스트 (Linked List)

예제) 4번째 숫자를 제거하세요.

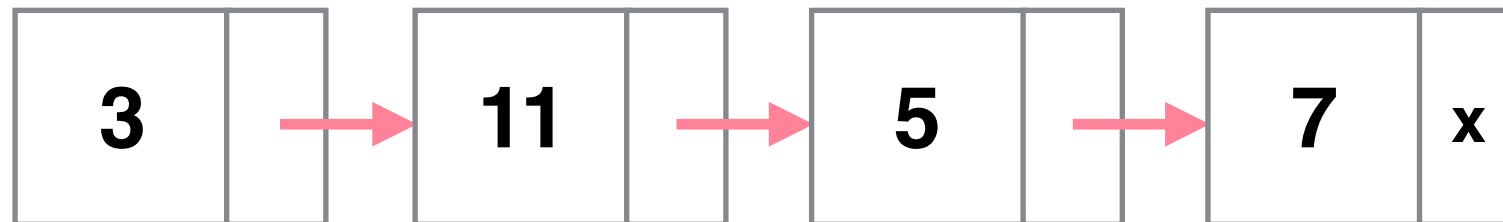
myLinkedList



# 링크드 리스트 (Linked List)

예제) 4번째 숫자를 제거하세요.

myLinkedList



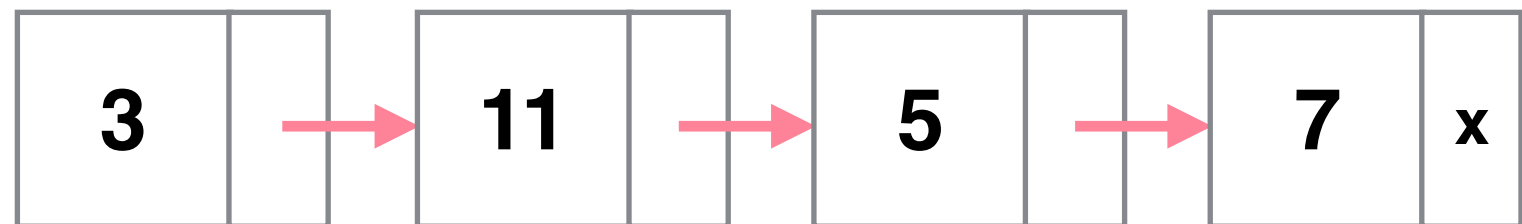
# 링크드 리스트 (Linked List)



**장점**

삽입/ 삭제가 빠르다

myLinkedList



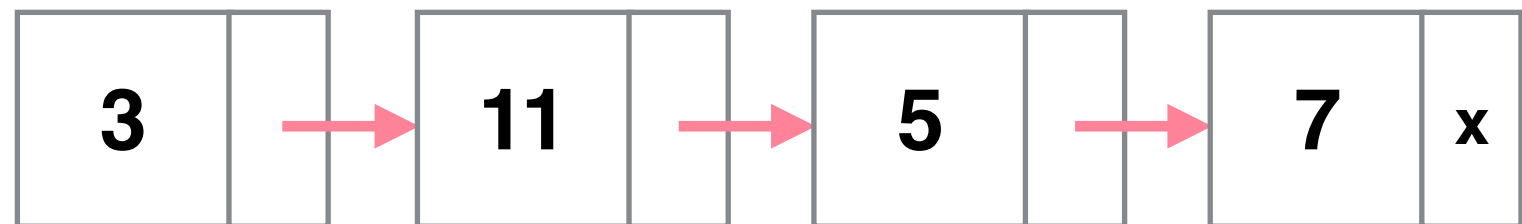
# 링크드 리스트 (Linked List)



**단점**

i 번째 원소를 알기가 쉽지 않다

myLinkedList



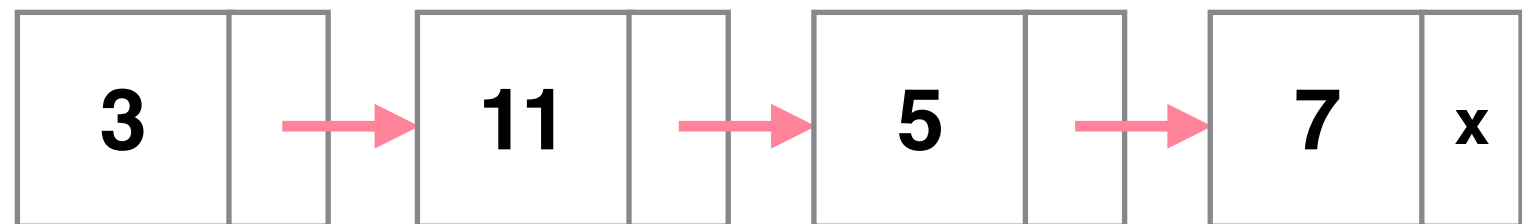
# 링크드 리스트 (Linked List)



**단점**

i 번째 원소를 알기가 쉽지 않다

myLinkedList



3번째 원소는?

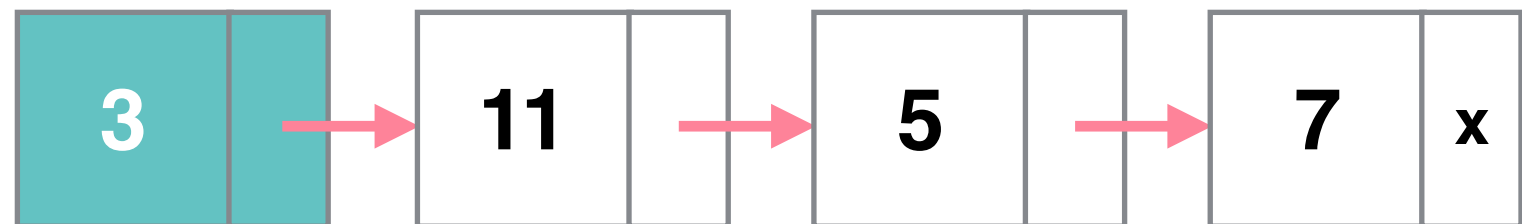
# 링크드 리스트 (Linked List)



**단점**

i 번째 원소를 알기가 쉽지 않다

myLinkedList



3번째 원소는?

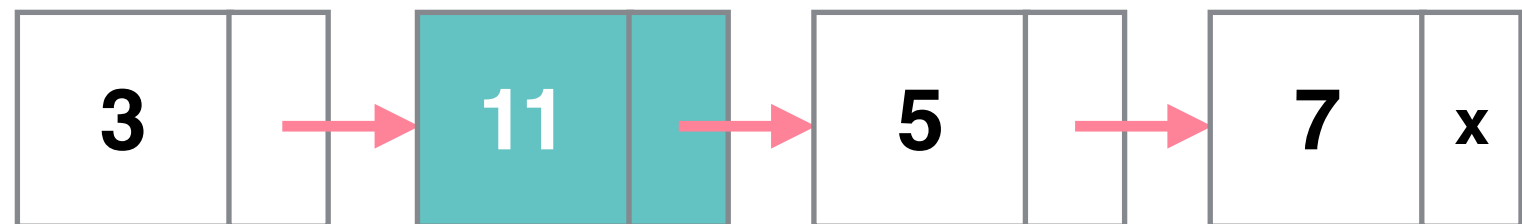
# 링크드 리스트 (Linked List)



**단점**

i 번째 원소를 알기가 쉽지 않다

myLinkedList



3번째 원소는?



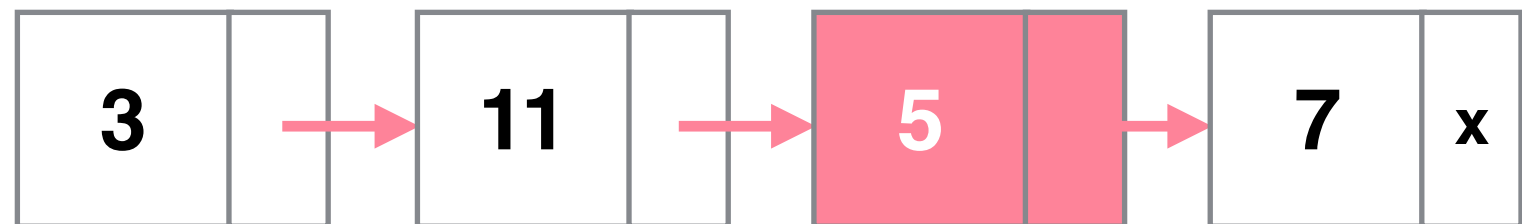
# 링크드 리스트 (Linked List)



**단점**

i 번째 원소를 알기가 쉽지 않다

myLinkedList



3번째 원소는?

# 요약

## 리스트 (List)

- **장점:**  $i$  번째 원소를 바로 알 수 있다
- **단점:** 원소의 추가 / 삭제가 까다롭다

## 링크드 리스트 (Linked List)

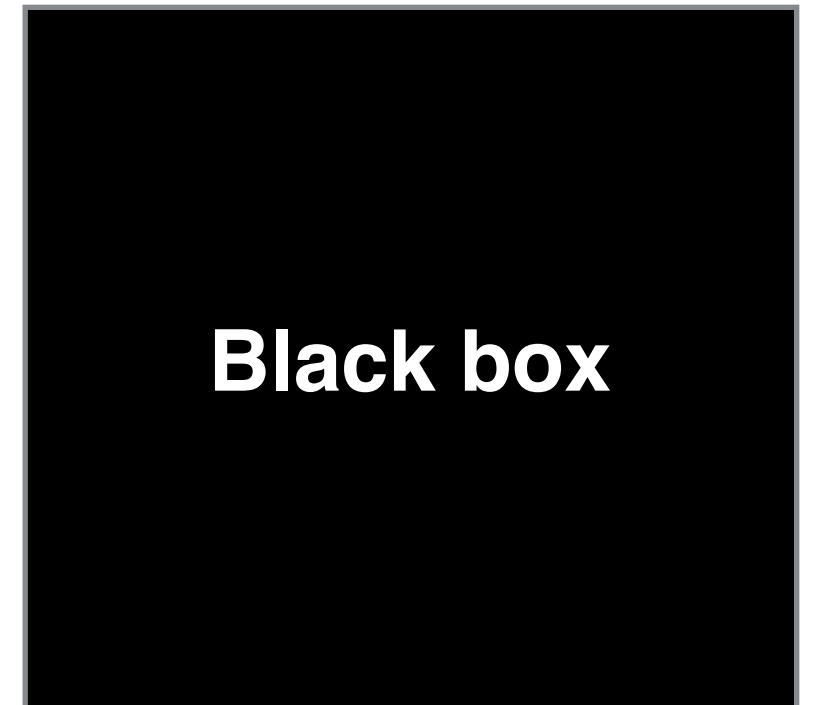
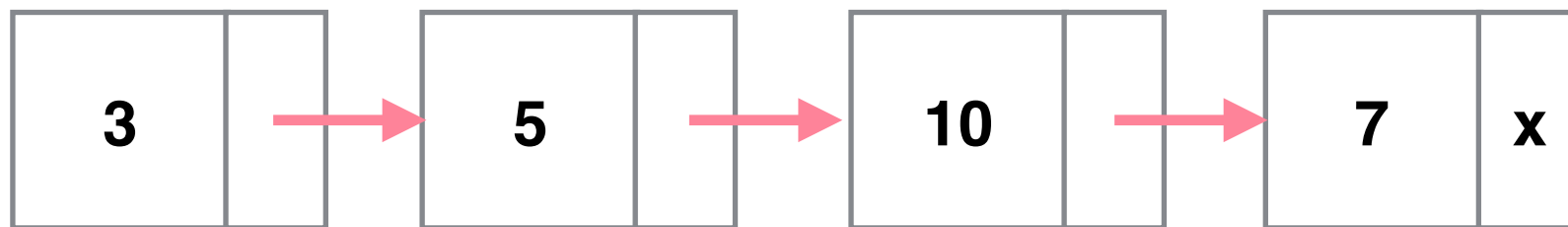
- **장점:** 원소의 추가 / 삭제가 간단하다
- **단점:**  $i$  번째 원소를 알기가 까다롭다

# 캡슐화: 자료구조 구현의 핵심

자료구조를 사용하는 사람은  
자료구조가 어떤 원리로 동작하는지 알 필요가 없다

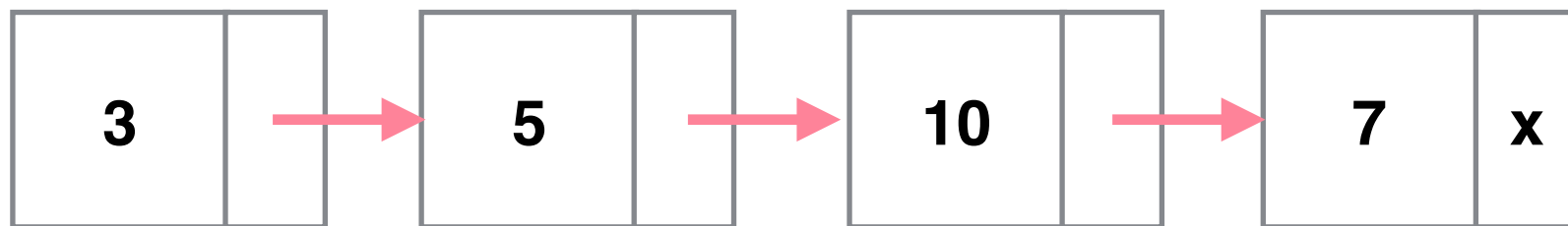
# 캡슐화 (Encapsulation)

자료구조를 사용하는 사람은  
자료구조가 어떻게 동작하는지 알 필요가 없다

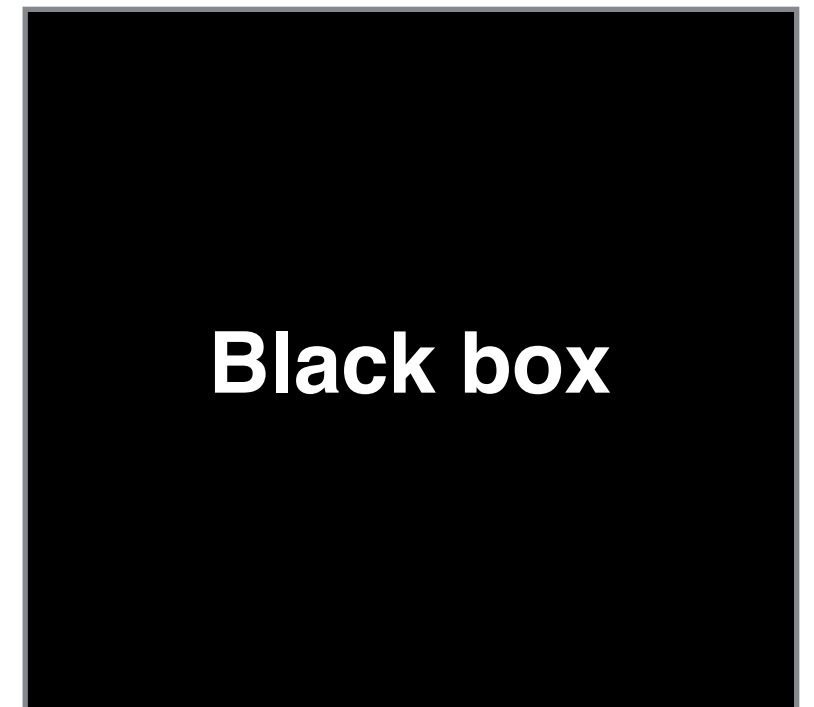


# 캡슐화 (Encapsulation)

자료구조를 사용하는 사람은  
자료구조가 어떻게 동작하는지 알 필요가 없다

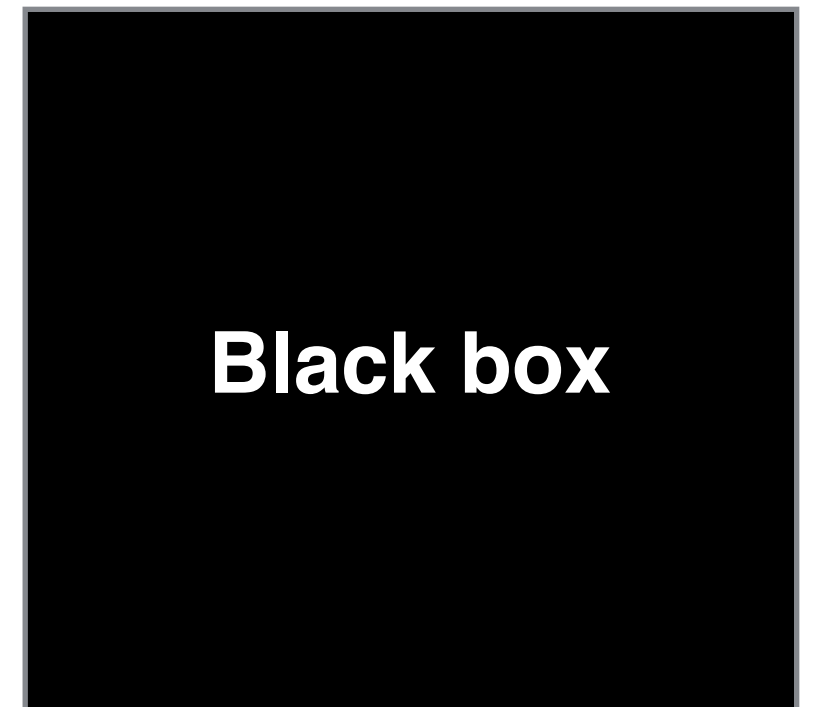
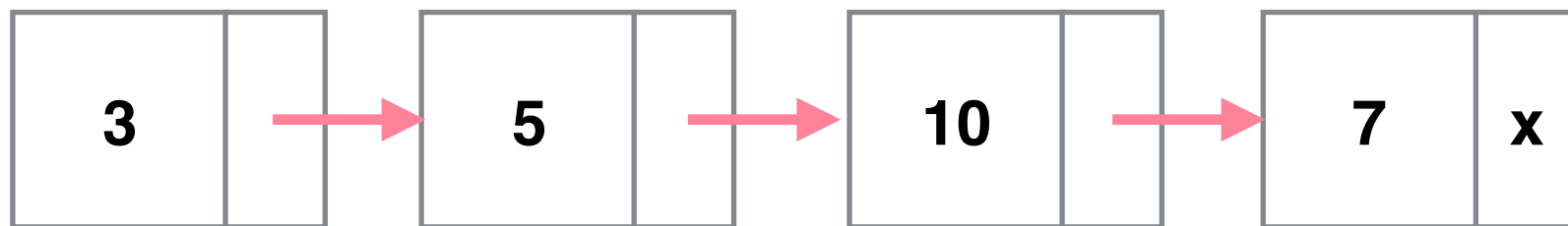


4번째 숫자 4를 넣어야지



# 캡슐화 (Encapsulation)

자료구조를 사용하는 사람은  
자료구조가 어떻게 동작하는지 알 필요가 없다



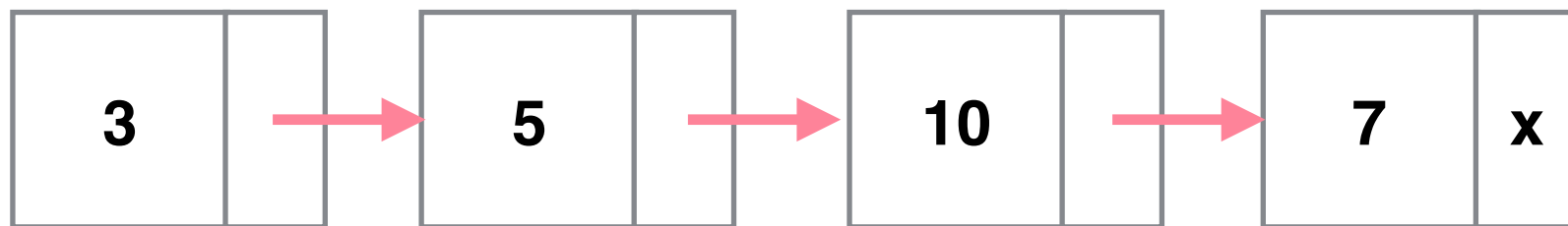
4번째 숫자 4를 넣어야지

“4번째에 숫자 4좀 넣어줘”

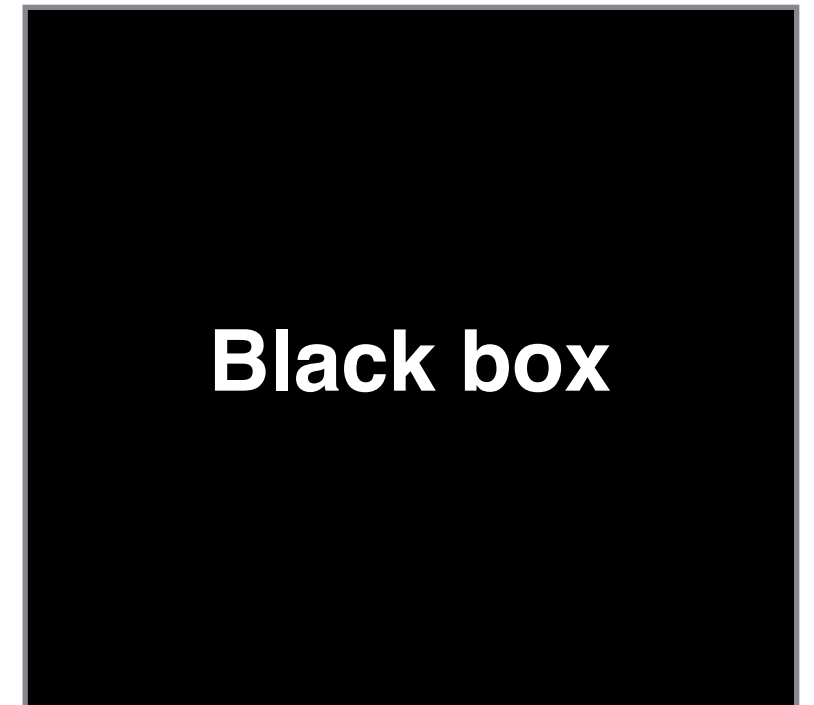


# 캡슐화 (Encapsulation)

자료구조를 사용하는 사람은  
자료구조가 어떻게 동작하는지 알 필요가 없다



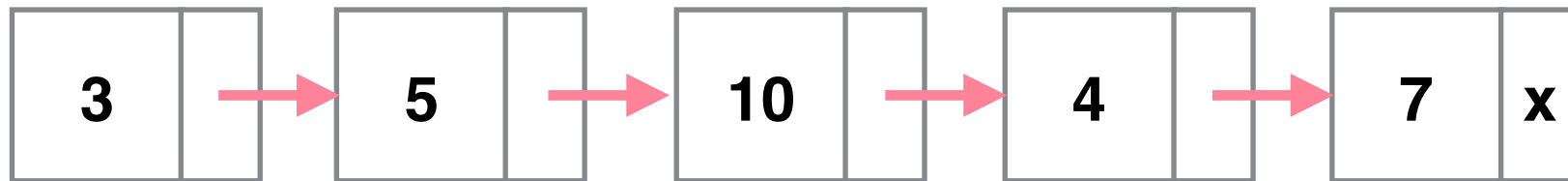
4번째 숫자 4를 넣어야지



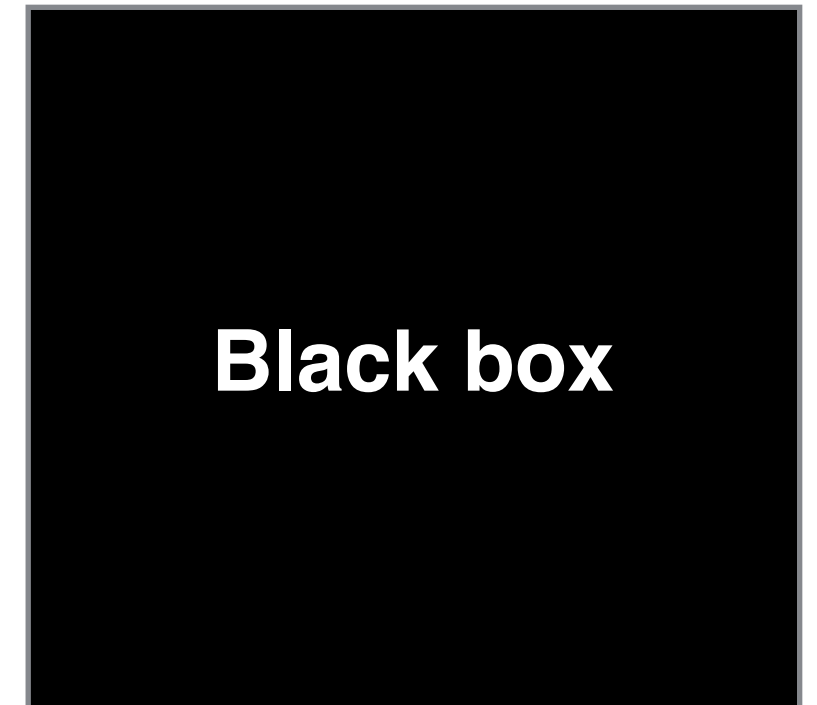
넣었습니다 :)

# 캡슐화 (Encapsulation)

자료구조를 사용하는 사람은  
자료구조가 어떻게 동작하는지 알 필요가 없다



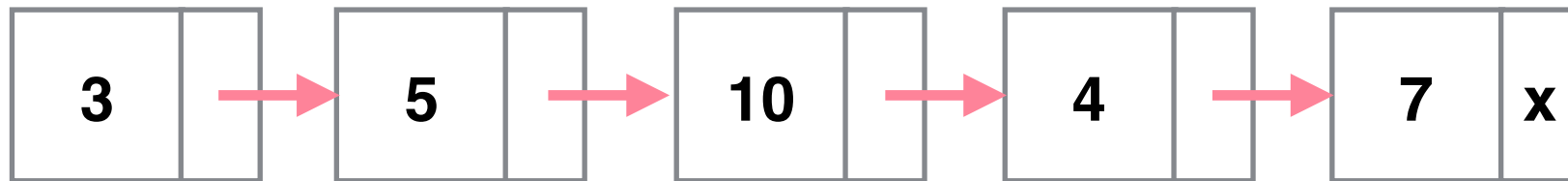
그럼 이제 되었겠군



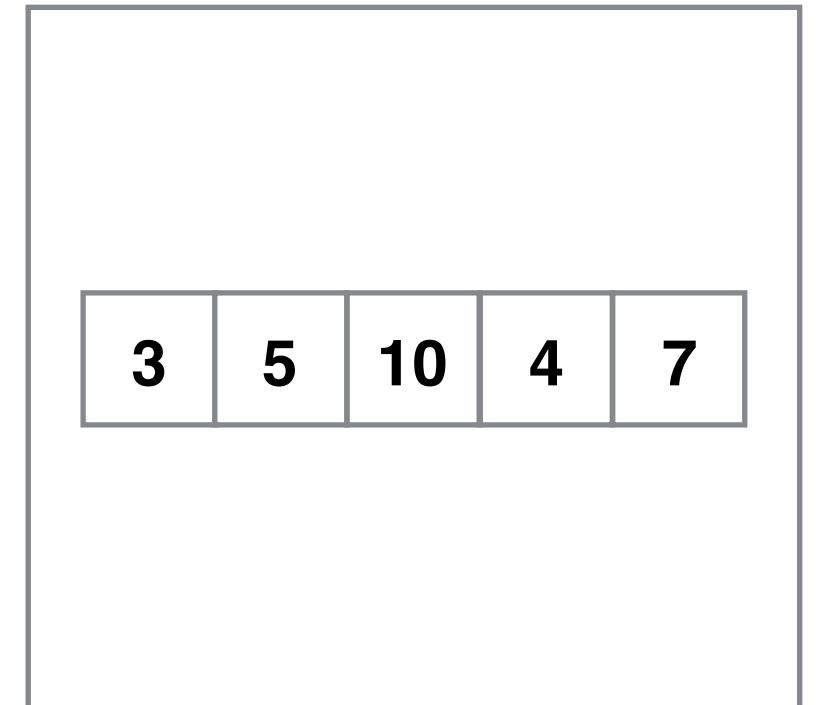


# 캡슐화 (Encapsulation)

자료구조를 사용하는 사람은  
자료구조가 어떻게 동작하는지 알 필요가 없다

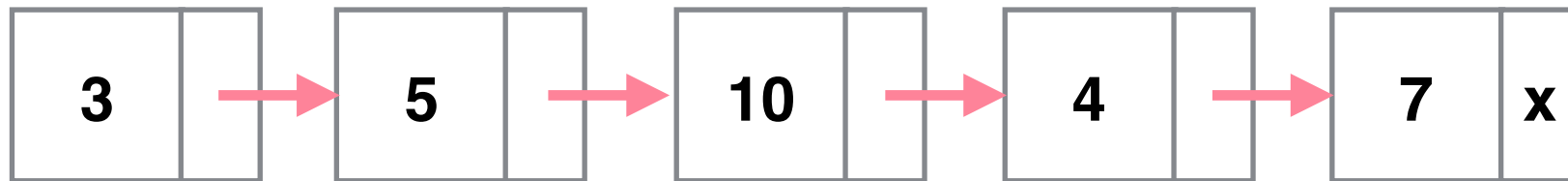


그럼 이제 되었겠군

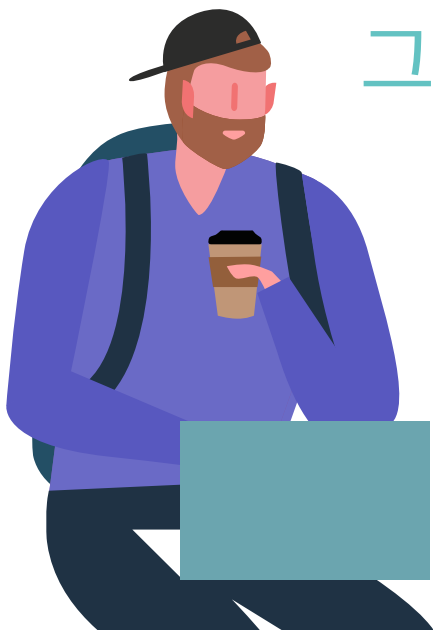


# 캡슐화 (Encapsulation)

자료구조를 사용하는 사람은  
자료구조가 어떻게 동작하는지 알 필요가 없다

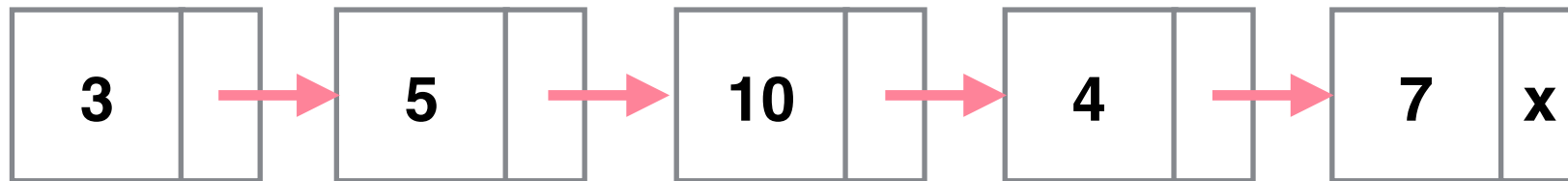


그럼 이제 되었겠군  
2번째 숫자를 빼자!



# 캡슐화 (Encapsulation)

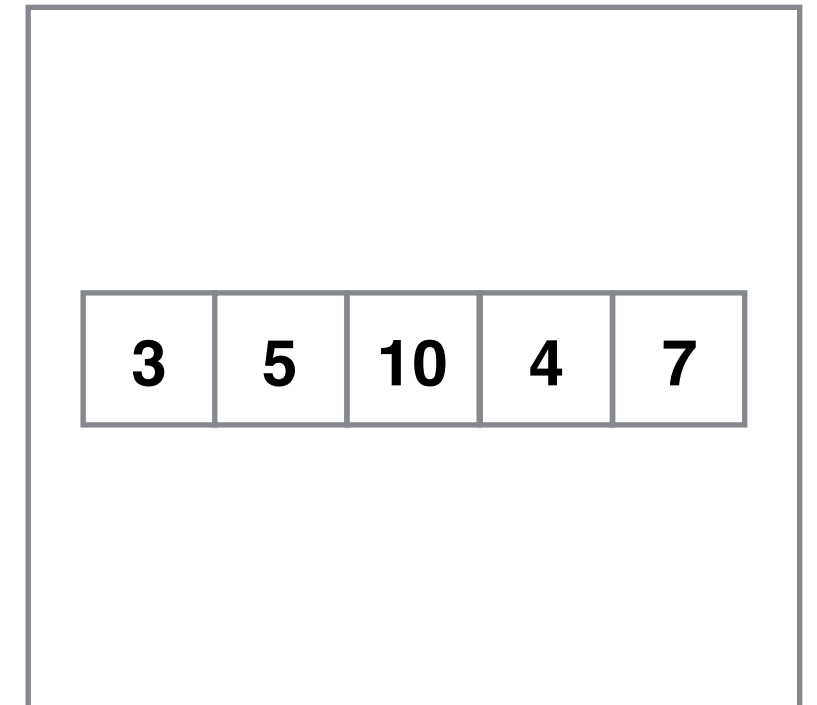
자료구조를 사용하는 사람은  
자료구조가 어떻게 동작하는지 알 필요가 없다



그럼 이제 되었겠군

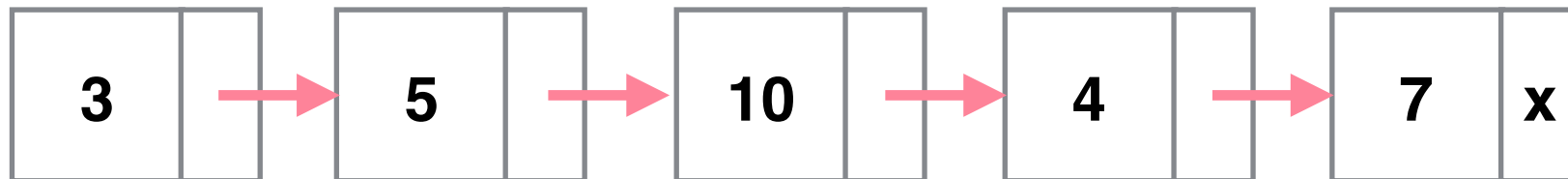
2번째 숫자를 빼자!

“2번째 숫자 좀 빼줘”



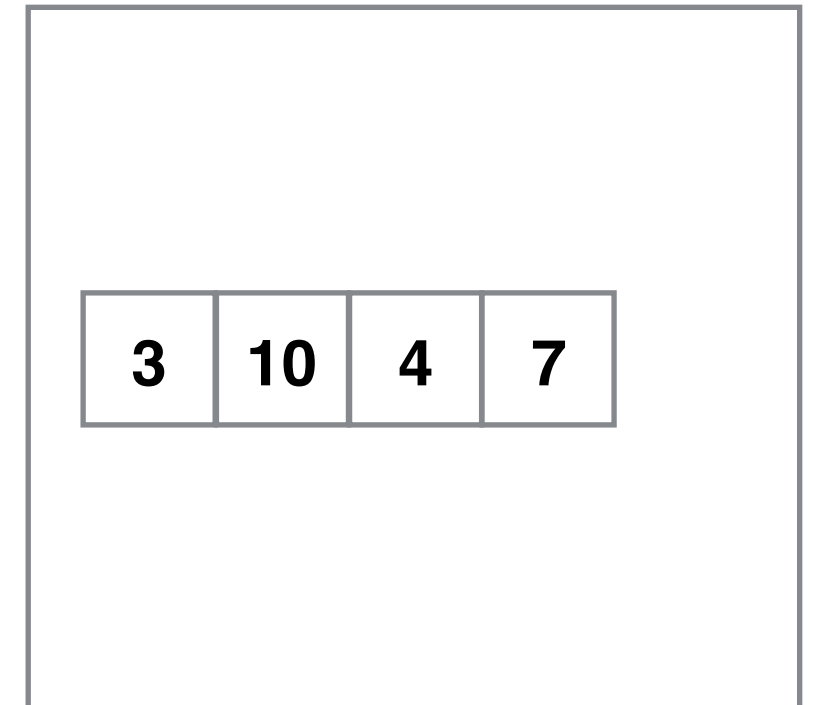
# 캡슐화 (Encapsulation)

자료구조를 사용하는 사람은  
자료구조가 어떻게 동작하는지 알 필요가 없다



그럼 이제 되었겠군

2번째 숫자를 빼자!

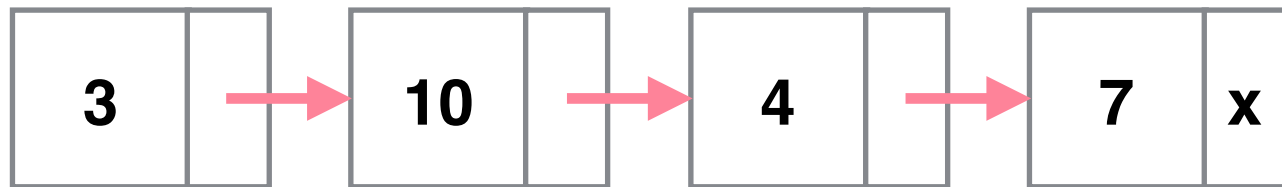


뺐습니다 :)

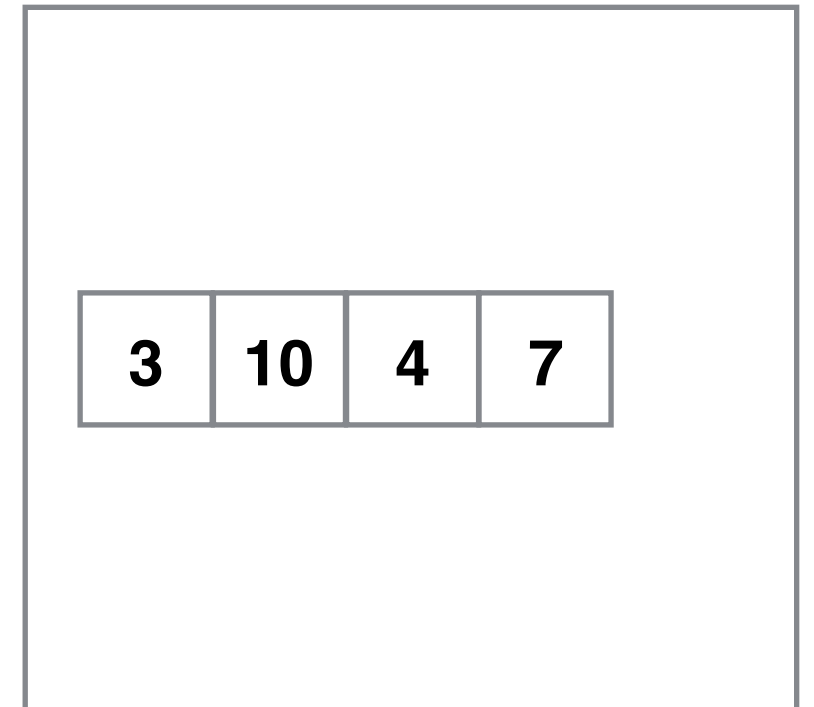


# 캡슐화 (Encapsulation)

자료구조를 사용하는 사람은  
자료구조가 어떻게 동작하는지 알 필요가 없다



ㅋㅋㅋ말잘듣넹



# 클래스 (Class): 캡슐화의 구현

클래스는 하나의 Black box를 구현할 때 사용한다

- 클래스
- 인스턴스

# 클래스 (Class): 캡슐화의 구현

클래스는 하나의 Black box를 구현할 때 사용한다

- 클래스 : “샴푸는 이렇게 생겼고, 이렇게 동작해야 해!”
- 인스턴스 : 실제 클래스를 바탕으로 만들어진 샴푸

# 구현 : [문제 1] 최댓값 기계





# 클래스 요약

## 사용자가 알아야 하는 것 (또는 사용자가 원하는 것)

- 자료구조에 숫자를 추가하기 위한 **명령**
- 자료구조에 있는 숫자를 제거하기 위한 **명령**
- 자료구조 내에 있는 숫자들 중 최대값을 구하기 위한 **명령**

## 제작하는 사람이 알아야 하는 것 (또는 구현해야 하는 것)

- 자료구조에 숫자를 추가하기 위한 **구현**
- 자료구조에 있는 숫자를 제거하기 위한 **구현**
- 자료구조 내에 있는 숫자들 중 최대값을 구하기 위한 **구현**

# 감사합니다!

신현규

E-mail : [hyungyu.sh@kaist.ac.kr](mailto:hyungyu.sh@kaist.ac.kr)

Kakao : yougatup

/\* elice \*/

## 문의 및 연락처

academy.elice.io

contact@elice.io

facebook.com/elice.io

[blog.naver.com/elicer](http://blog.naver.com/elicer)