

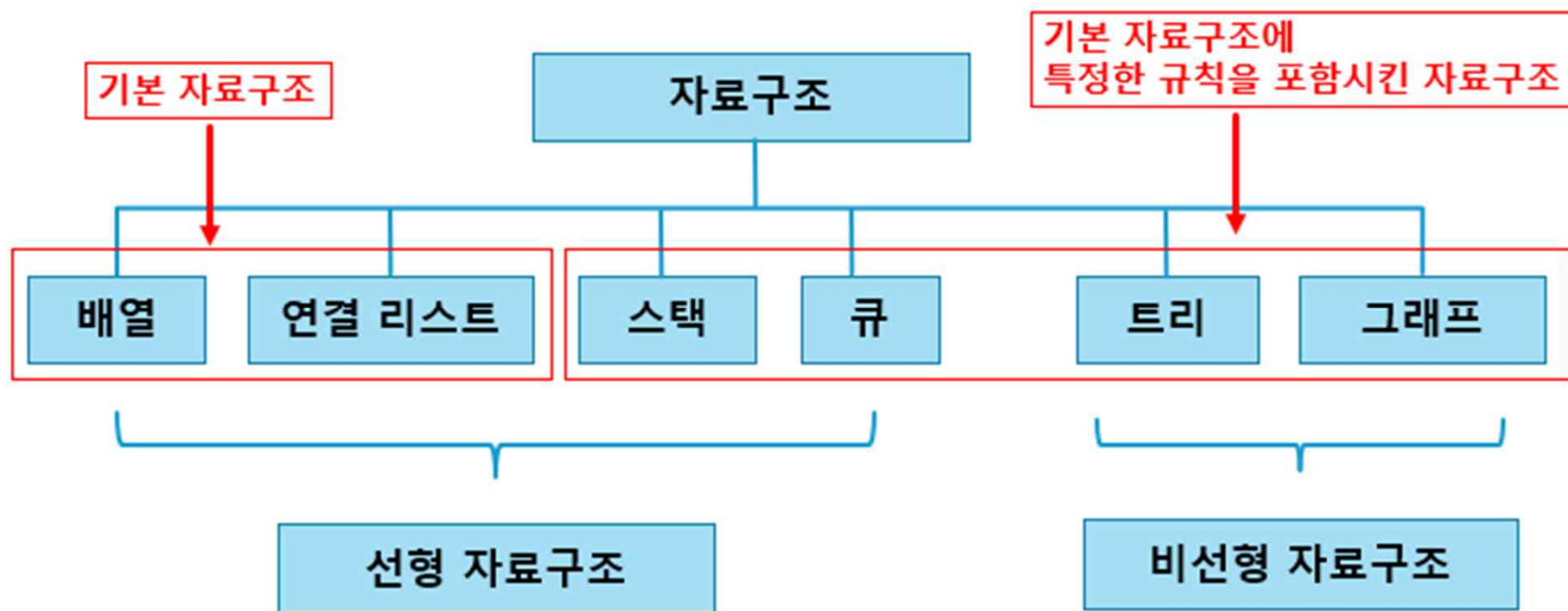
03. 기본 자료구조와 확장형 자료구조 1

배열

연결리스트



자료구조





자료구조



- 배열로 구현한 스택

→ 여러 개의 정수형 데이터를 스택 규칙에 따라 배열로 저장

- 배열로 구현한 큐

- 배열과 연결 리스트로 구현한 트리

- 배열과 연결 리스트로 구현한 그래프

- 연결 리스트로 구현한 스택

→ 여러 개의 정수형 데이터를 스택 규칙에 따라 연결 리스트로 저장

- 연결 리스트로 구현한 큐



자료구조

□ 선형 자료구조

- ▣ 순서가 정해져 있는 자료구조
- ▣ 데이터를 정해진 자료형으로 순서대로 저장
- ▣ 저장된 여러 개의 데이터 중에서 특정 데이터를 순차적으로 접근

□ 비선형 자료구조

- ▣ 순서가 정해져 있지 않은 자료구조
- ▣ 데이터를 정해진 자료형으로 순서와 상관없이 저장
- ▣ 기본 자료구조를 다양하게 적용하여 구현
- ▣ 저장된 여러 개의 데이터 중에서 특정 데이터에 접근하려면 저장된 규칙을 알아야 함



기본 자료구조

- 기본 자료구조
 - ▣ 배열, 연결 리스트
 - ▣ 확장형 자료구조(스택, 큐, 트리, 그래프)는 기본 자료구조를 이용하여 데이터를 메모리에 저장



배열

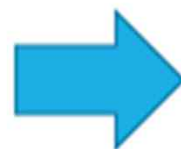
□ 배열

- 동일한 타입의 원소들이 연속적인 메모리 공간에 할당되어 각 항목이 하나의 원소에 저장되는 기본적인 자료구조
- 데이터들은 순차적으로 저장
- 특정 원소에 접근할 때는 배열의 인덱스를 이용하여 한 번에 접근
- 인덱스는 배열의 첫 번째 원소를 0으로 하고 1씩 증가
- 배열은 개념이 쉽고 사용하기 편하지만 메모리 효율성이 낮음
- 배열은 미리 정해진 크기의 메모리 공간을 할당 받은 뒤 사용
- 새 항목을 배열 중간에 삽입 혹은 삭제 시 뒤따르는 항목들을 한 칸 씩 뒤로 또는 앞으로 이동시켜야 하는 부담이 따름
- 파이썬 리스트는 스마트한 배열



배열

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]
일	월	화	수	목	금	토



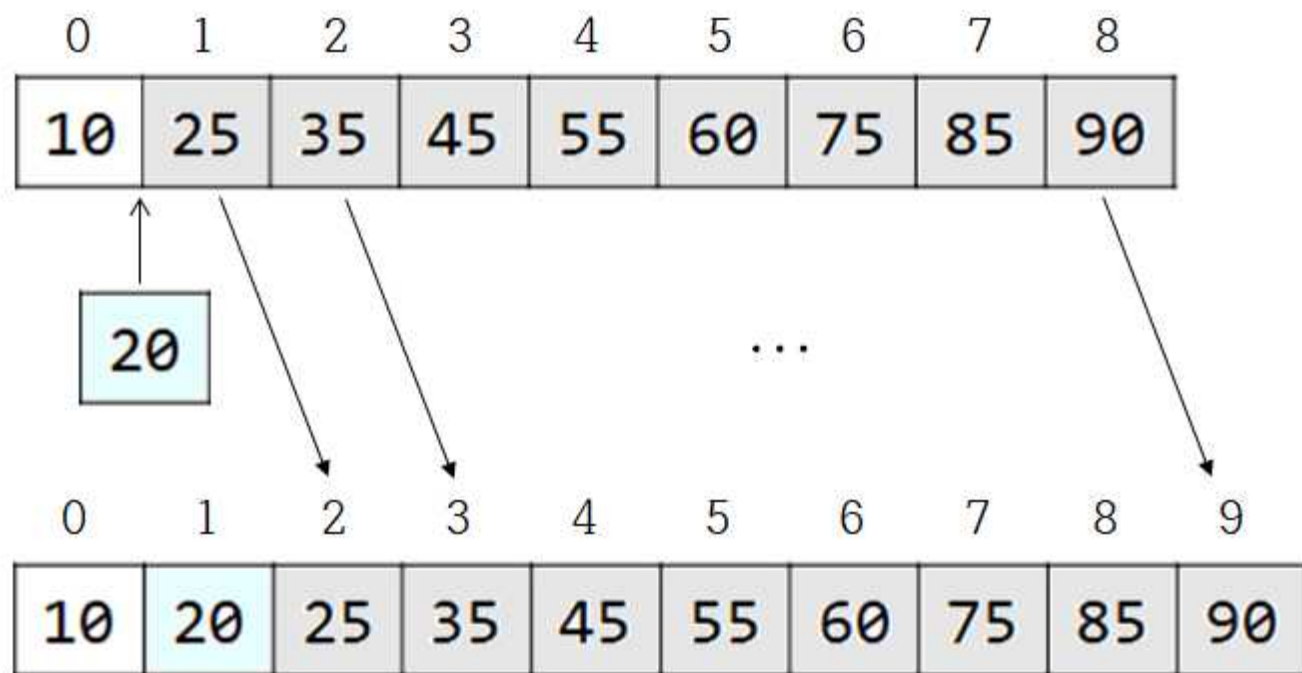
A

일	A[0]
월	A[1]
화	A[2]
수	A[3]
목	A[4]
금	A[5]
토	A[6]



배열

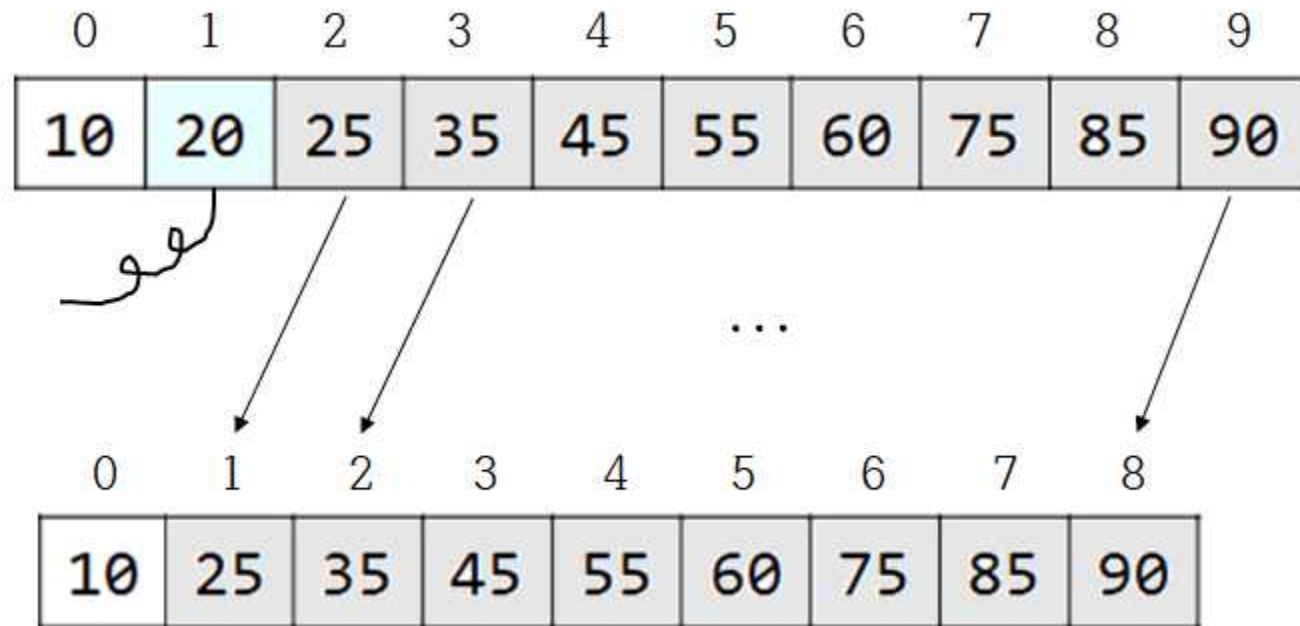
- 배열(파이썬 리스트)에서는 삽입 시 항목들이 이동





배열

- 배열(파이썬 리스트)에서는 삭제 시 항목들이 이동

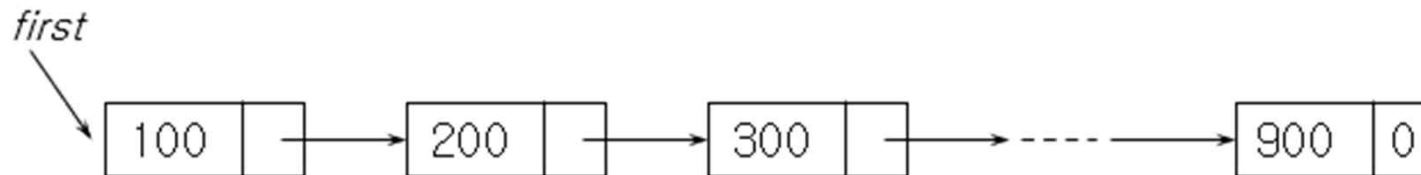




연결 리스트

□ 연결 리스트(Linked List)

- 동적 메모리 할당을 이용해 리스트를 구현하는 간단한 형태의 자료구조
- 동적 메모리 할당을 받아 노드를 만들고, 노드는 레퍼런스를 이용하여 다음 노드를 가리키도록 만들어 노드들을 한 줄로 연결시킴
 - 레퍼런스는 다음 노드의 메모리 주소를 가지며 포인터라고도 함



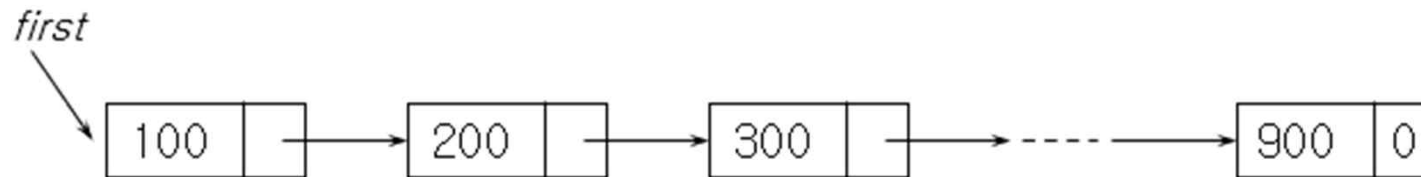
- 연결 리스트는 단순 연결 리스트, 이중 연결 리스트, 원형 연결 리스트가 있음



연결 리스트

□ 연결 리스트(Linked List)

- ▣ 배열의 경우 최초에 배열의 크기를 예측하여 결정하므로 대부분의 경우 배열에 빈 공간이 있으나, 연결 리스트는 빈 공간이 없어 메모리 효율성이 높음
- ▣ 단순 연결 리스트에서의 탐색은 항상 첫 노드부터 원하는 노드를 찾을 때까지 순차 탐색해야 함
 - 데이터 탐색 수행 시간이 느림

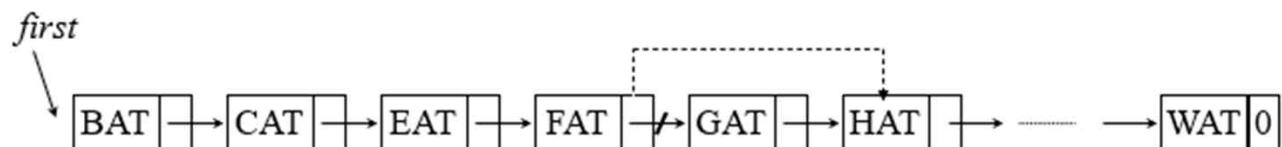
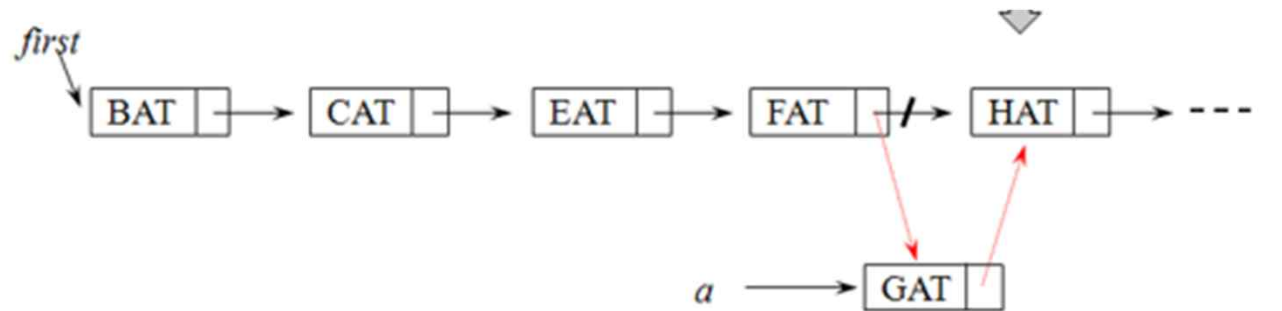




연결 리스트

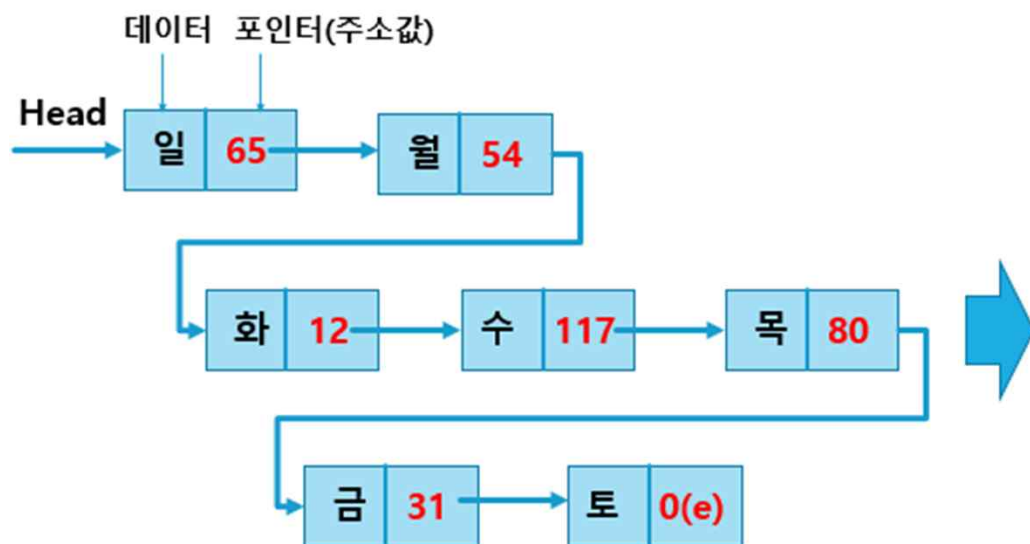
□ 연결 리스트(Linked List)

- 단순 연결 리스트에서는 삽입이나 삭제 시 항목(노드)들의 이동이 필요 없음
 - 삽입 시 노드를 만들고 레퍼런스만 변경하면 됨
 - 삭제 시 단지 레퍼런스만 변경하면 됨
 - 배열 구조에서는 항목의 이동이 수반됨





연결 리스트



	...
12	수 117
	...
31	토 0
	...
Head	70
	...
54	화 12
	...
65	월 54
	...
70	일 65
	...
80	금 31
	...
117	목 80



גג
גג