

이 세상에서 찾아보느

자료구조의 원리

02주차

번호 붙이기 - 배열과 파이썬 리스트

1차시

번호를 붙이는 자료구조, 배열

» 메모리를 사용하는 구조

- ['a', 'b', 'c', 'd', 'e'] + ['f', 'g', 'h', 'i']

[illegible]

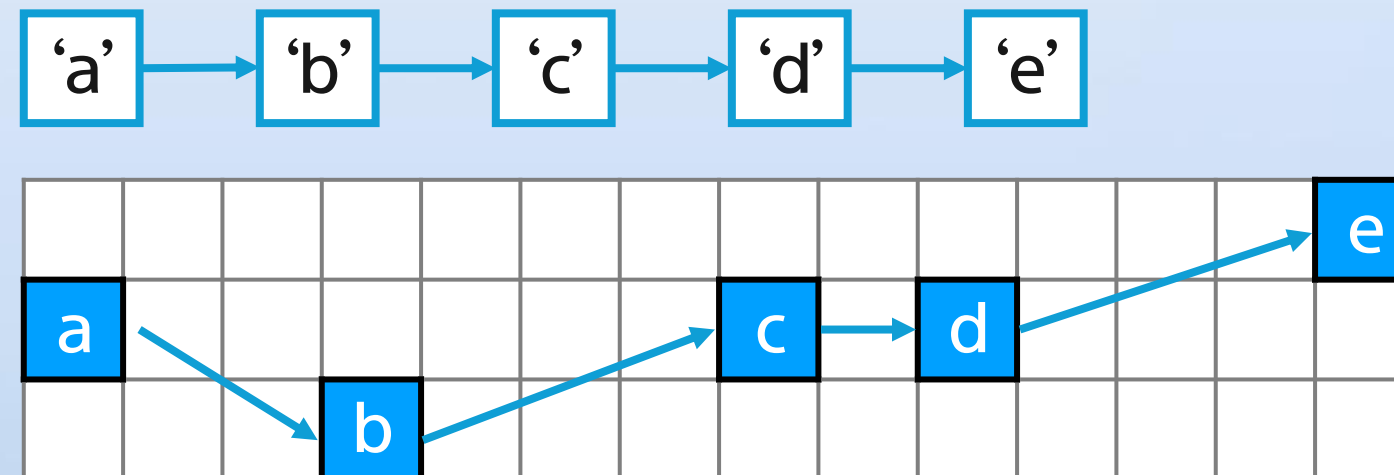
['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i']

[illegible]

● 들어가기 전에...

» 메모리를 사용하는 구조

- 연결 리스트(Linked List)
 - 연속되지 않아도 되므로 공간의 제약이 덜 함
 - 서로 연결된 관계를 알아야 함



학습목표

- » 번호 붙이기의 간편성과 편리성에 대해 설명할 수 있다.
- » 배열의 번호를 의미하는 인덱스란 무엇인지 설명할 수 있다.
- » 배열이 메모리에 저장되는 구조를 설명할 수 있다.

학습내용

- » 일상에서 사용하는 배열
- » 배열의 구조, 동작, 특징
- » 번호 붙이기
- » 파이썬 코드보기
- » 번호 붙이는 자료구조
- » 배열의 활용

○ 일상에서 사용하는 배열

키보드 자판 배열

상점의 물건 배열

도서관 책의 배열



📖 배열

- 같은 종류의 물건을 차례대로 나열해 두는 것
- 각각의 위치에는 번호를 매겨, 원하는 물건을 쉽게 찾을 수 있음

○ 번호 붙이기

» 번호 붙이기의 간편성

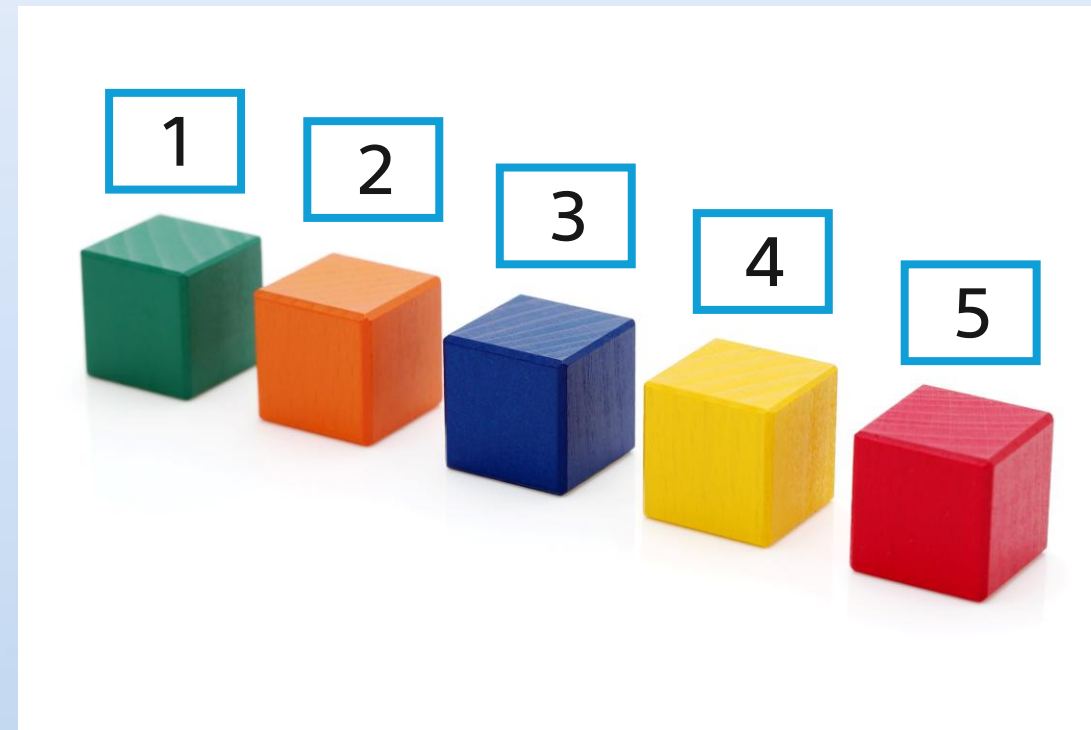
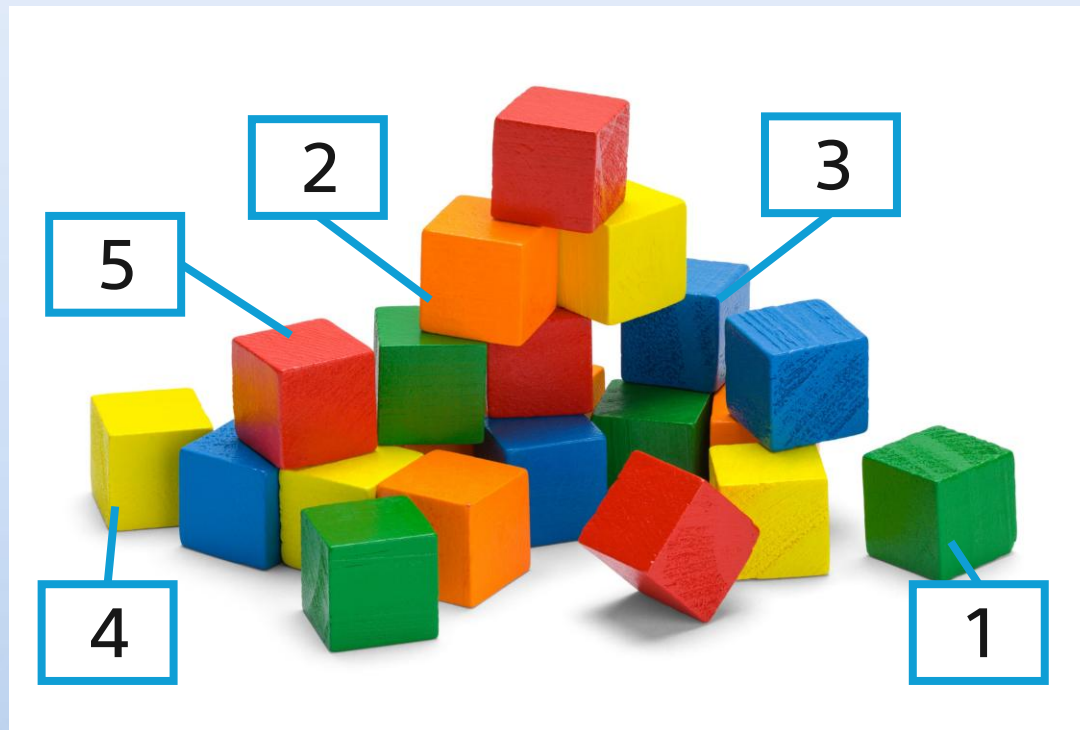
- 자료나 물건의 종류에 상관 없음
- 번호로 순서도나 중요도를 표시할 수 있음



○ 번호 붙이기

» 번호 붙이기의 편리성

- 번호를 지정하면 자료를 바로 꺼내어 볼 수 있음
- 번호 순서대로 나열하기에 편리함



○ 번호 붙이기

» 예: 책꽂이

- 번호가 붙어 있는 각 칸에는 책이 하나씩 들어가 있음
- 번호를 통해서 몇 번째 칸에 어떤 책이 있는지 쉽게 알 수 있음



✓ 책 정리하기 (자료 저장하기)

- 책꽂이의 각 칸에 책을 한 권씩 넣음
- 1번 칸에는 수학 책, 2번 칸에는 과학 책, 3번 칸에는 영어 책을 넣음

✓ 책 찾기 (자료 읽기)

- 번호를 통해 원하는 책을 쉽게 찾을 수 있음
- 2번 칸에 있는 책을 찾으려면 과학 책을 찾을 수 있음

● 번호 붙이는 자료구조

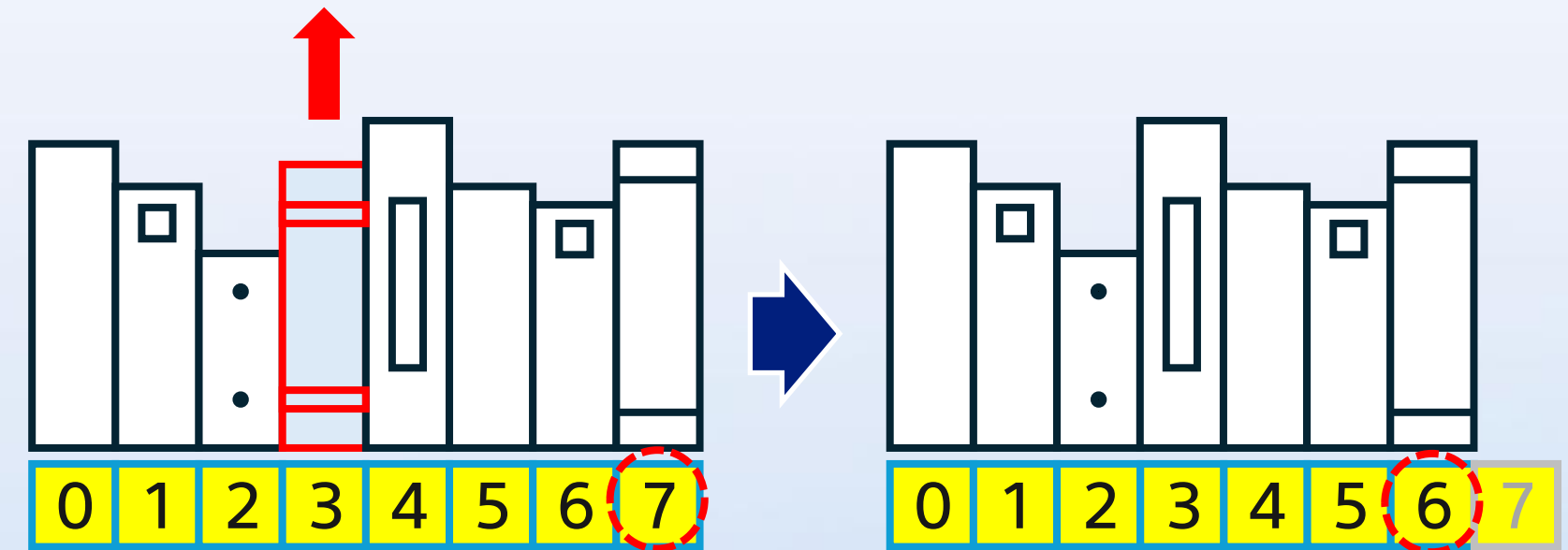
» 배열(Array)

- 일렬로 나열된 자료구조
- 자료에 번호를 붙임

➔ 이 번호를 '인덱스'라 함

0부터 시작

- 연속으로 저장함



마지막 인덱스 + 1 = 전체 자료의 개수

- ✓ 책장에서 책을 꺼낸 후 빈 공간을 만들지 않음
 - 마지막 위치의 번호로 전체 책의 개수를 알 수 있음
 - 빈 공간을 낭비하지 않음

» 메모리에서 배열의 특징

- ➡ 배열의 첫 번째 원소가 메모리의 어떤 위치에 있고, 두 번째 원소는 그 다음 위치에 있음

[illegible]

○ 번호 붙이는 자료구조

» 메모리에서 배열의 특징

- 빠르게 접근

- 배열의 각 원소는 번호(인덱스)를 사용하여 한 번에 찾아냄

- ➡ 책장에서 몇 번째 칸에 있는 책을 바로 찾는 것과 같음

배열의 구조, 동작, 특징

배열의 구조

- 배열의 이름: **book**
 - 전체 자료를 구분함
- 배열의 원소: 수학책, 과학책, 영어책
 - 개별 자료(값), 원소
 - 인덱스: **[0] [1] [2]**
 - 배열의 크기: 3
 - 원소의 개수
 - 마지막의 인덱스보다 1개 더 많음(배열 크기: n, 마지막 인덱스 = n-1)

[수학책, 과학책, 영어책]

book	수학책	과학책	영어책
인덱스	[0]	[1]	[2]

[가연, 철희, 영수, 연지]

person	가연	철희	영수	연지
인덱스	[0]	[1]	[2]	[3]

배열의 구조, 동작, 특징

>> 배열의 구조

- 배열의 개별 원소

- 배열이름[인덱스]

- 수학책 → book[0]

- 과학책 → book[1]

- 영어책 → book[2]

[수학책, 과학책, 영어책]

book	수학책	과학책	영어책
인덱스	[0]	[1]	[2]

- 배열의 원소는 직접 다룰 수 없음

- 인덱스를 지정하거나 읽거나 변경, 삭제

- 예 book[1]을 컴퓨터책으로 바꾸어라

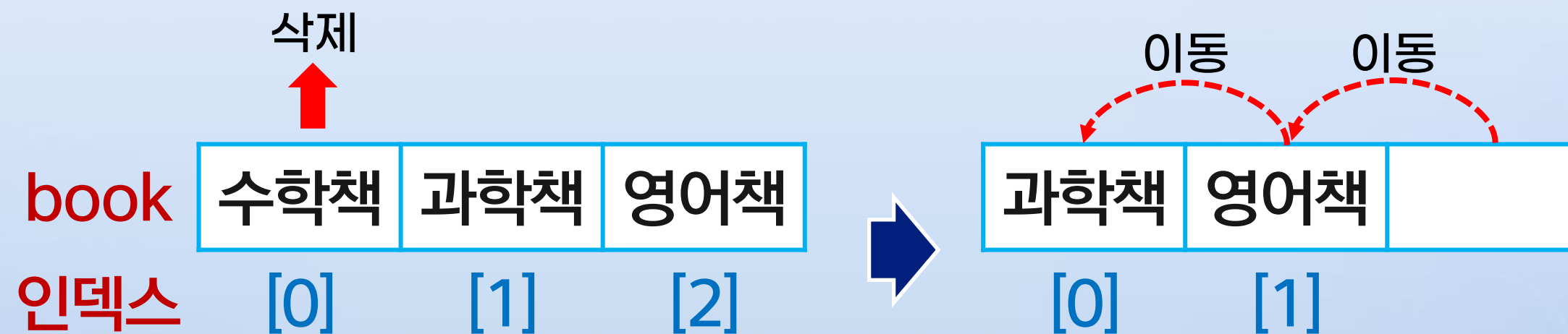
- book[0]을 삭제하라

배열의 구조, 동작, 특징

배열의 동작

자료의 삭제

- **인덱스**를 통해서 삭제할 원소를 지정함
- 삭제 후에도 **연속적인 공간**에 저장되도록 원소를 이동함



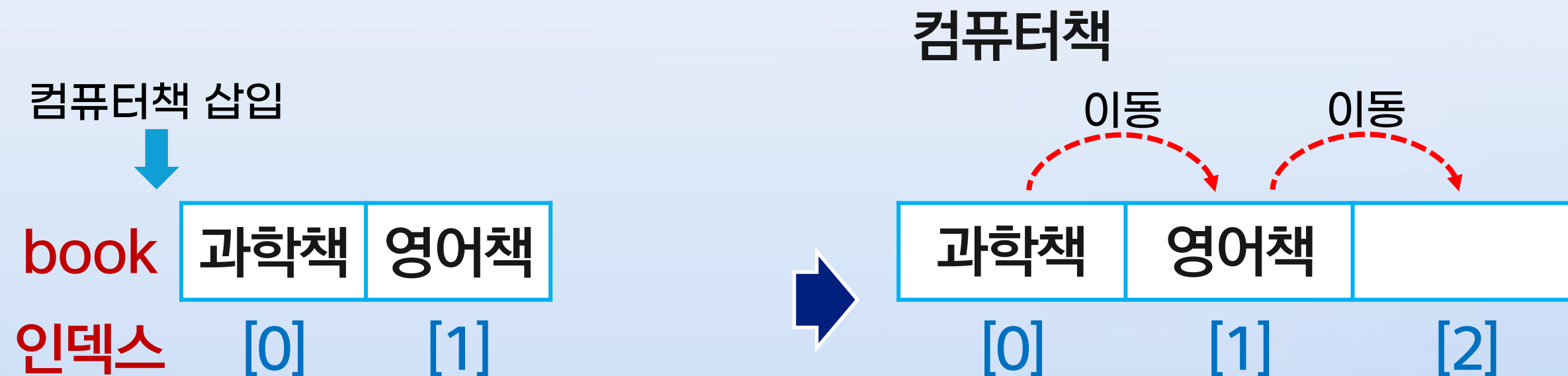
➡ 첫번째 원소 삭제 → 나머지 원소들의 이동 → 원소가 2개 남음

➡ 빈 공간이 없어짐

배열의 구조, 동작, 특징

>> 배열의 동작

- 새로운 자료의 삽입



- 원소들이 차례로 이동함(덮어쓰기 방지)
- 빈 공간에 새로운 값을 삽입함

배열의 구조, 동작, 특징

» 배열의 특징

- 배열은 선형구조
 - list, linear list, ordered list
 - ‘순서’에 의미가 있음(번호를 붙이며, 번호를 인덱스라고 부름)
 - 원소들은 인덱스로 구분함
- 배열의 제약점
 - 순서를 맞추기 위해서 삽입 및 삭제 시 원소들을 이동시켜야 함
 - 크기에 따라서 메모리의 연속된 공간을 확보해야 함

배열의 구조, 동작, 특징

» 배열의 특징

- 배열의 장점
 - 번호를 이용한 간단한 원리로 구현함
 - 원하는 자료에 번호를 통해서 한번에 접근함

파이썬 코드 보기

```
book = [10, 20, 30]
```

- 리스트는 대괄호로 생성함
- 각 원소는 쉼표(,)로 구분함

```
print(book)
```

- 리스트의 이름을 통해서 원소를 출력해 볼 수 있음

```
print(book[2])
```

- 인덱스를 이용해서 특정 원소를 출력해 볼 수 있음

book	10	20	30
인덱스	[0]	[1]	[2]

파이썬 코드 보기

Run

Markdown

파이썬 리스트 기본 실습

1 book = [10, 20, 30] #리스트를 만든다

2 print(book) #리스트 book을 출력해 본다

[10, 20, 30]

1 print(book[2]) #인덱스 2의 원소를 출력한다.

30

book	10	20	30
인덱스	[0]	[1]	[2]

파이썬 코드 보기

```
del book[0]
```

- del 키워드로 특정 원소를 삭제함

```
book.insert(0, 40)
```

- insert는 리스트에 새로운 원소를 삽입하는 기능
- insert(삽입할 위치의 인덱스, 원소의 값)

파이썬 코드 보기

↑

↓

▶ Run

■

↺

▶▶

Code

▼

1

book = [10, 20, 30] #리스트를 만든다

2

print(book) #리스트 book을 출력해 본다

[10, 20, 30]

1

print(book[2]) #인덱스 2의 원소를 출력한다.

30

1

del book[0]

2

print(book)

3

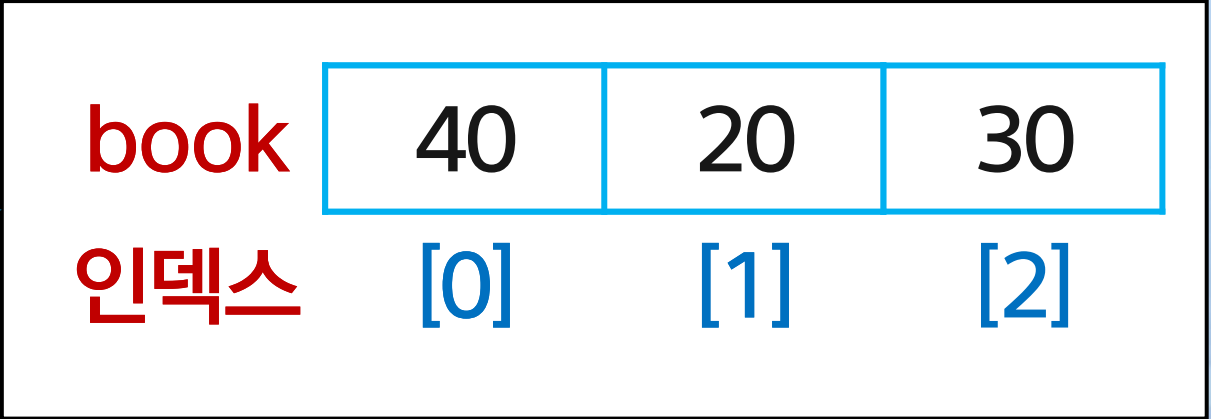
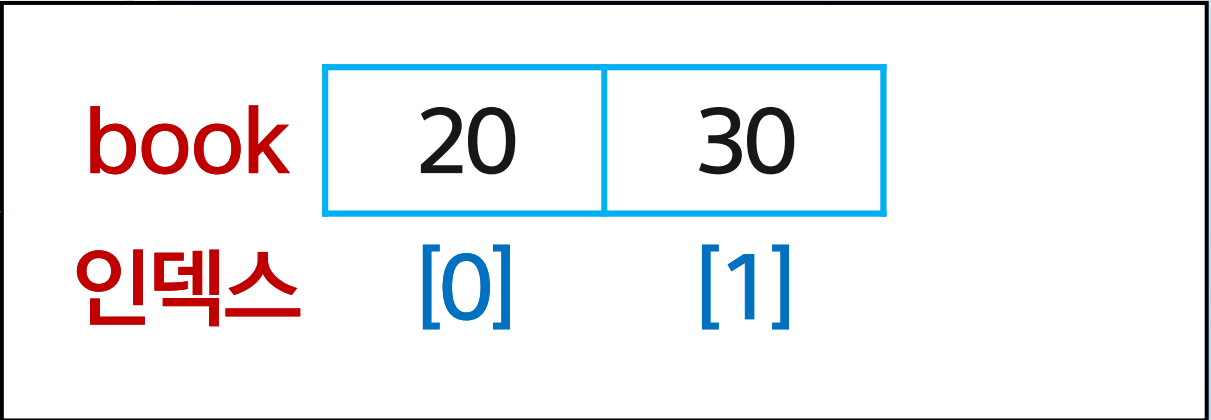
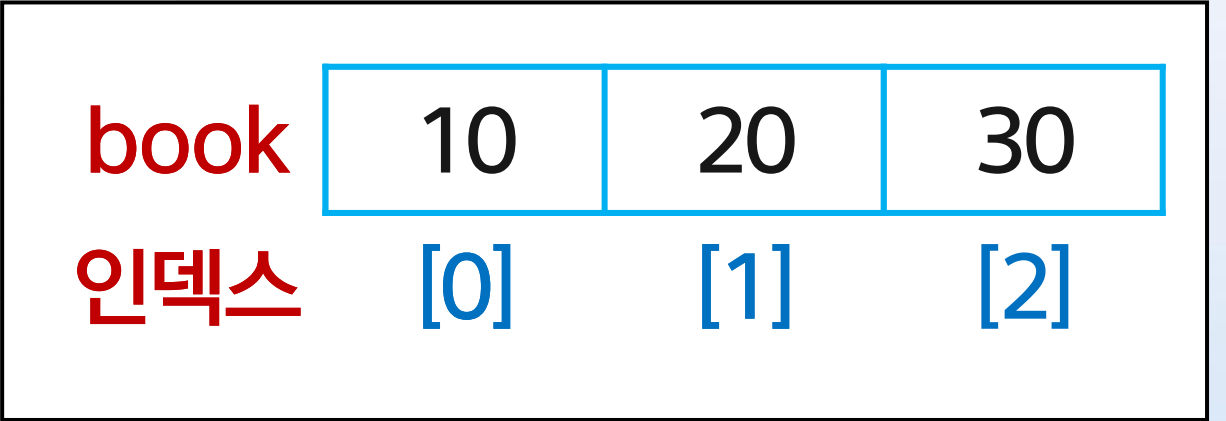
book.insert(0, 40)

4

print(book)

[20, 30]

[40, 20, 30]



배열의 활용

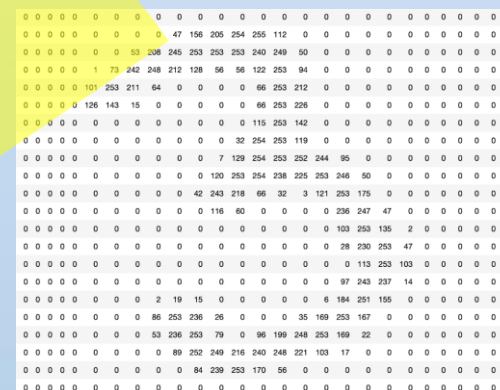
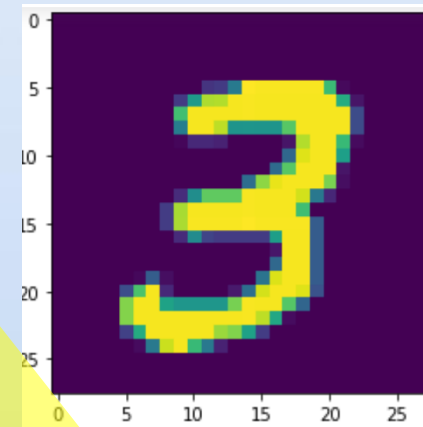
1 배열은 자료를 모아서 저장

– 간단하고 효율적인 자료구조

2 행렬이나 이미지 표현

— 다차원 배열

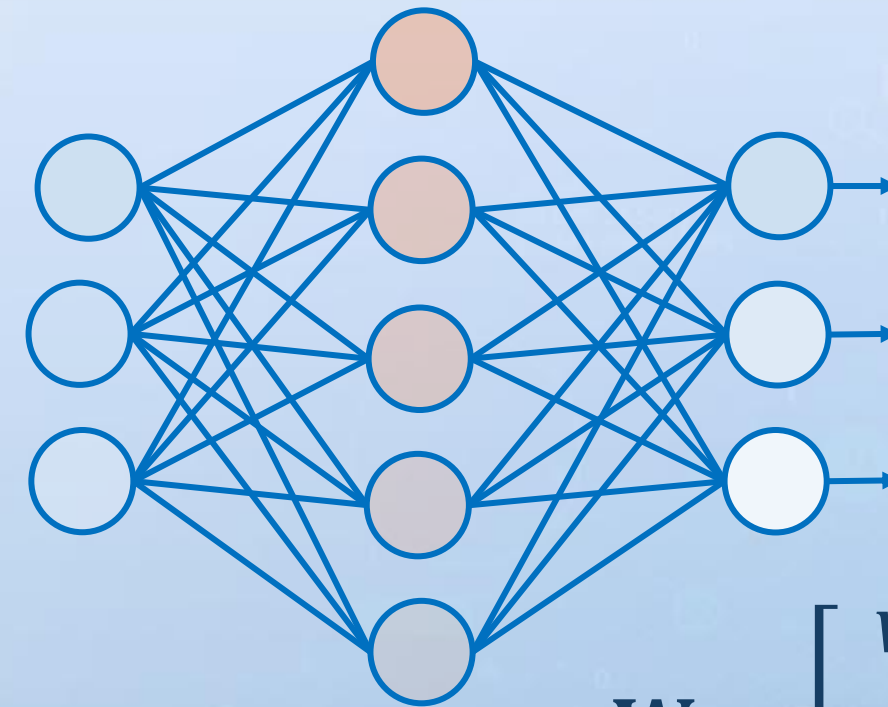
0	0	0	0	0	0
0	0	47	156	205	254
53	208	245	253	253	253
242	248	212	128	56	56
211	64	0	0	0	0
15	0	0	0	0	0



3 인공신경망 데이터를 다차원 배열로 표현

– 입력값, 출력값

- 가중치, ...



$$\mathbf{y} = [y_1, y_2, y_3]$$

$$\mathbf{W} = \begin{bmatrix} w_{11} & w_{21} & w_{31} \dots \\ w_{12} & w_{22} & w_{33} \dots \\ \dots & \dots & \dots \end{bmatrix}$$

배열의 활용

4 연속된 메모리 할당

– 배열의 요소의 위치를 간단한 연산으로 계산할 수 있음

→ 모든 원소에 대한 액세스가 상수 시간에 이루어질 수 있음을 의미함

$$W = \begin{bmatrix} w_{11} & w_{21} & w_{31} & \dots \\ w_{12} & w_{22} & w_{32} & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix}$$



w_{11}	w_{21}	w_{31}	w_{12}	w_{22}	w_{32}	...
[0]	[1]	[2]	[3]	[4]	[5]	...

+3

+3

+3

○ 학습정리

💡 배열의 개요

- 같은 자료형의 자료들을 일렬로 배열하는 구조
 - 개별 자료는 인덱스로 구분함
 - 파이썬 리스트로 구현하면 편리함
 - 서로 다른 자료형도 배열 형태로 구현 가능

배열 이름	‘자료’	‘구조’
인덱스	[0]	[1]