

이 세상에서 찾아보느

자료구조의 원리

02주차

번호 붙이기 - 배열과 파이썬 리스트

2차시

파이썬 리스트와 배열

학습목표

» 파이썬 리스트 실습을 통해서 배열 구조를 설명할 수 있다.

학습내용

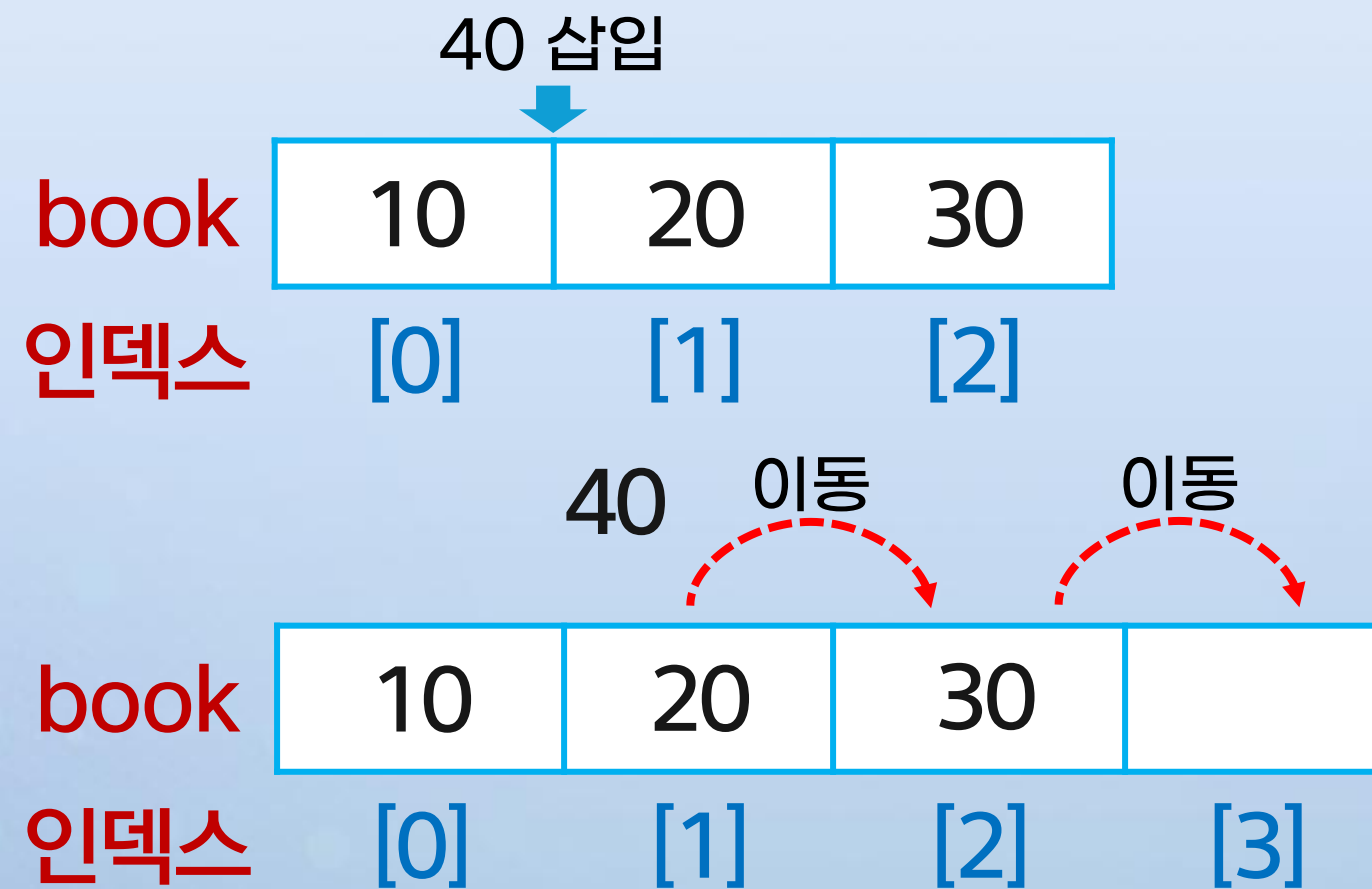
- » 파이썬 리스트
- » 리스트의 삽입과 삭제
- » 리스트 원소 읽기
- » 파이썬 리스트 정리하기
- » 파이썬 리스트 고려 사항

파이썬 리스트

» 리스트

- 인덱스를 사용
- 연속적인 공간에 자료의 순서를 유지하는 구조 → 배열

✓ 원소의 삽입 과정



```
for i = 3 to 2
    book[i] = book[i-1]
    i = i-1
book[i-1] = 40
```

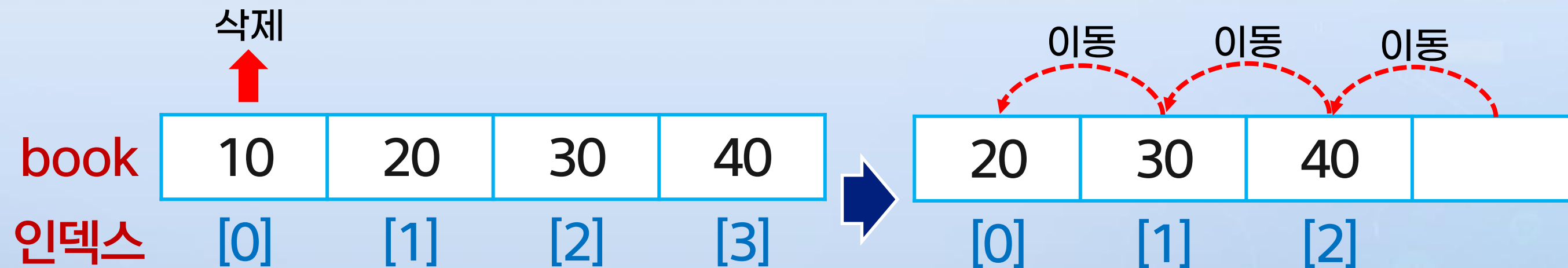
이동

삽입

○ 파이썬 리스트

» 파이썬 리스트의 기능

- 원소의 삭제, 삽입, 변경, 읽기 등
- ‘리스트 이름.함수’의 형식으로 사용
 - `book.pop(0)`



○ 파이썬 리스트

» 리스트 원소를 삽입하고 삭제하는 함수

- **insert(인덱스, 자료값), pop(인덱스)**

numbers

1	2	3	4	5
[0]	[1]	[2]	[3]	[4]

```
numbers.insert(0, -1)
```

-1	1	2	3	4	5
[0]	[1]	[2]	[3]	[4]	[5]

```
numbers.pop(0)
```

1	2	3	4	5
[0]	[1]	[2]	[3]	[4]

리스트의 삽입과 삭제

```
numbers = [1,2,3,4,5]
print(numbers)
numbers.insert(0, -1) #인덱스 0에 -1 삽입
print(numbers)
numbers.pop(0) #인덱스 0의 원소 삭제
print(numbers)
numbers.insert(4, 10)
print(numbers)
```

numbers

1	2	3	4	5
[0]	[1]	[2]	[3]	[4]

-1	1	2	3	4	5
[0]	[1]	[2]	[3]	[4]	[5]

1	2	3	4	5
[0]	[1]	[2]	[3]	[4]

1	2	3	4	10	5
[0]	[1]	[2]	[3]	[4]	[5]

리스트의 삽입과 삭제

```
intList = [10, 12, 23, -4, 0]
```

#정수로 구성된 리스트

```
strList = ["Hello"]
```

#문자열 하나로 구성된 리스트

```
charList = ['h', 'e', 'l', 'l', 'o']
```

#문자들로 구성된 리스트

```
myList = [0, 'a', 'high', -0.9]
```

#여러 자료형의 원소로 구성된 리스트

```
print(intList)
```

```
print(strList)
```

```
print(charList)
```

```
print(myList)
```

```
[10, 12, 23, -4, 0]
```

```
['Hello']
```

```
['h', 'e', 'l', 'l', 'o']
```

```
[0, 'a', 'high', -0.9]
```

리스트의 삽입과 삭제

» 마지막에 새로운 원소를 추가하는 함수: `append()`

```
strList = ["Hello"]  
print(strList)  
strList.append("world")  
print(strList)  
strList.append("!")  
print(strList)
```

```
['Hello']  
['Hello', 'world']  
['Hello', 'world', '!']
```

'hello'

[0]

'hello'

'world'

[0]

[1]

'hello'

'world'

'!'

[0]

[1]

[2]

리스트 원소 읽기

» 슬라이싱

- 리스트이름[시작:끝:간격] 형식(인덱스는 음수, 양수 다 가능)

```
print(intList[1:3])
```

#인덱스 1부터 3 이전까지, 즉 인덱스1, 2의 원소

[-5]	[-4]	[-3]	[-2]	[-1]
10	12	23	-4	0
[0]	[1]	[2]	[3]	[4]

```
print(intList[:])
```

#처음부터 끝까지

[-5]	[-4]	[-3]	[-2]	[-1]
10	12	23	-4	0
[0]	[1]	[2]	[3]	[4]

리스트 원소 읽기: 슬라이싱

» 슬라이싱

- 리스트이름[시작:끝:간격] 형식(인덱스는 음수, 양수 다 가능)

```
print(charList[ : :2])
#처음부터 끝까지 인덱스 +=2로 이동
```

[-5]	[-4]	[-3]	[-2]	[-1]
'h'	'e'	'l'	'l'	'o'
[0]	[1]	[2]	[3]	[4]

```
print(myList [-4:4:3])
#인덱스 -4에서 3까지 인덱스 +=3로 이동
```

[-4]	[-3]	[-2]	[-1]
0	'a'	'high'	-0.9
[0]	[1]	[2]	[3]

○ 파이썬 리스트 정리해 보기

1 리스트를 다루는 여러가지 함수를 제공함

- insert - append
- pop - list
- - len
- - ...

A	B	C
[0]	[1]	[2]

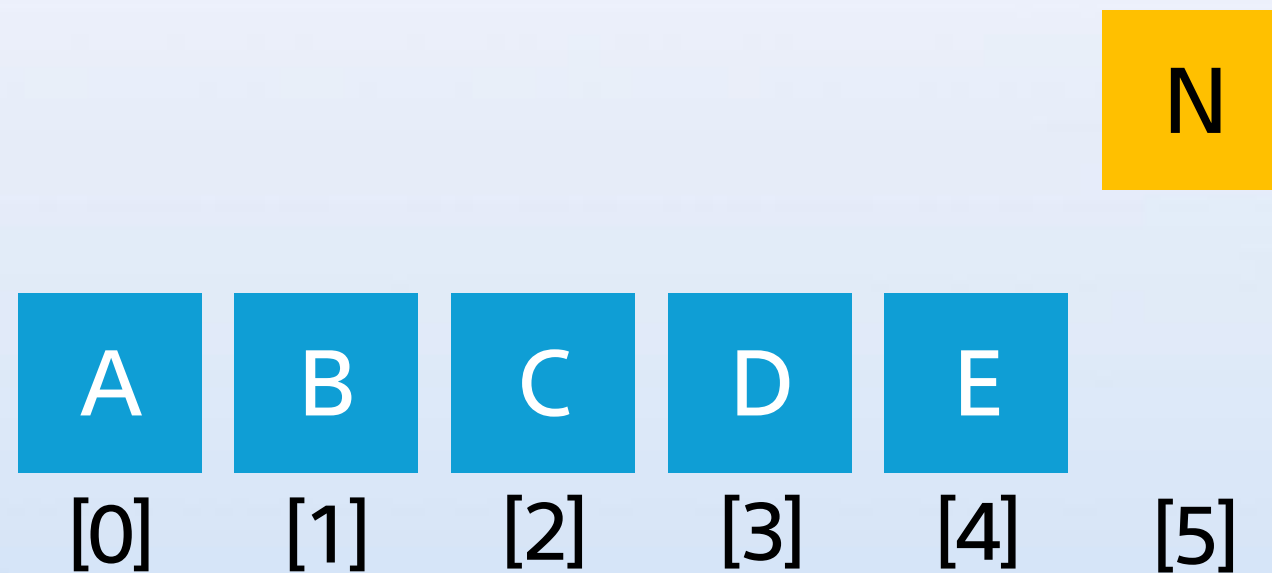
2 인덱스는 번호임

3 원소에 직접 접근할 수 있음

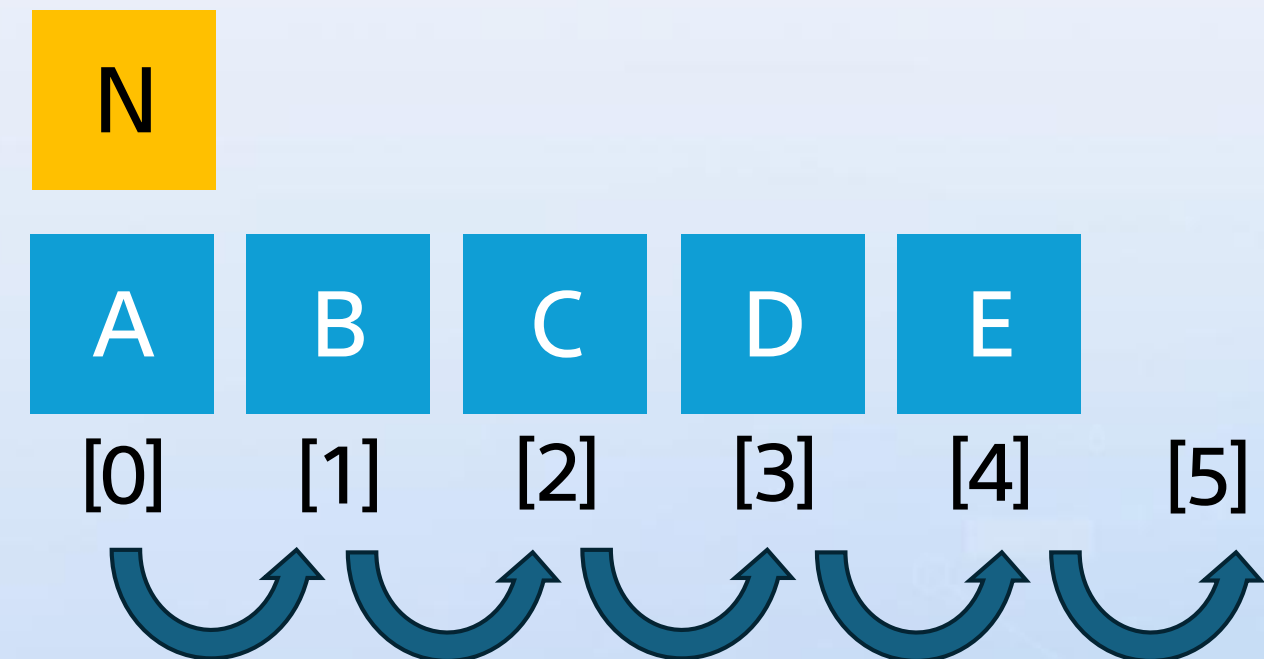
- 슬라이싱
- 음수, 양수 인덱스

파이썬 리스트 고려 사항

» append(N): $O(1)$



» insert(0,N): $O(n)$



○ 파이썬 리스트 고려 사항

» 파이썬 리스트의 효율적 사용

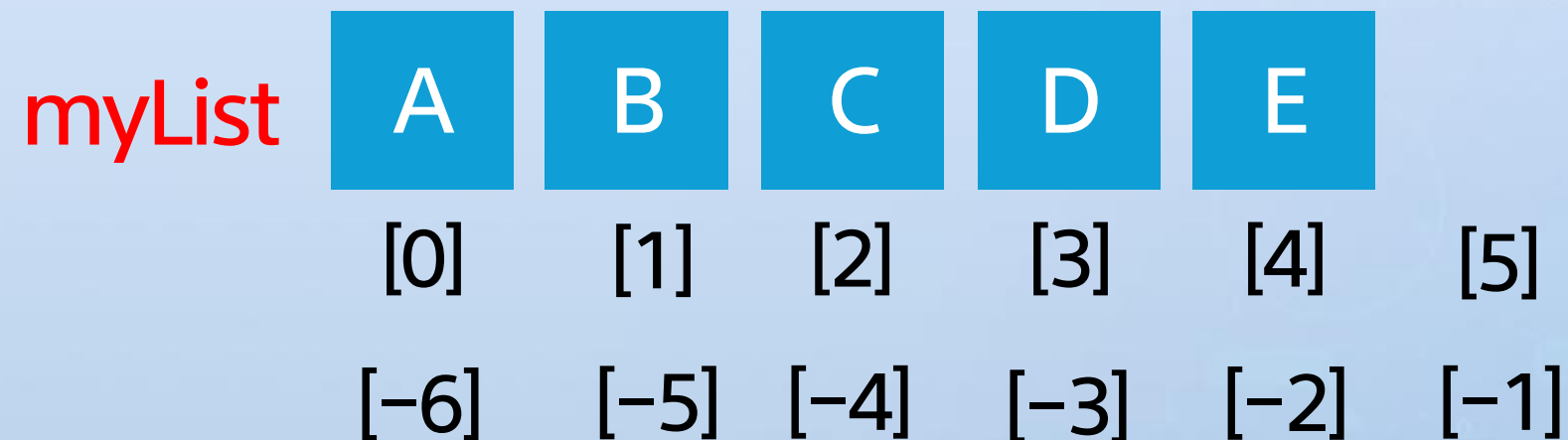
- 후단(마지막 위치) 삽입이나 삭제
- 중간이나 전단(처음 위치) 삽입이나 삭제는 비효율적임

– `myList.append(N)`

– `myList.pop(-1)`

➡ -1은 마지막 위치

N



○ 학습정리

💡 파이썬 리스트

- 배열 구조를 구현하기에 편리함
 - 인덱스를 사용
 - 배열 요소에 삽입 및 삭제, 수정 등의 기능 제공

