

리스트 (List)

1. 배열 vs 연결리스트

2. 파이썬의 리스트

3. 리스트 컴프리헨션



1. 배열 vs 연결리스트

1) 배열 (Array)

여러 데이터들이 연속된 메모리 공간에 저장되어 있는 자료구조

```
int arr[5] = {70, 80, 20, 100, 90};
```

C언어에서의 배열 선언

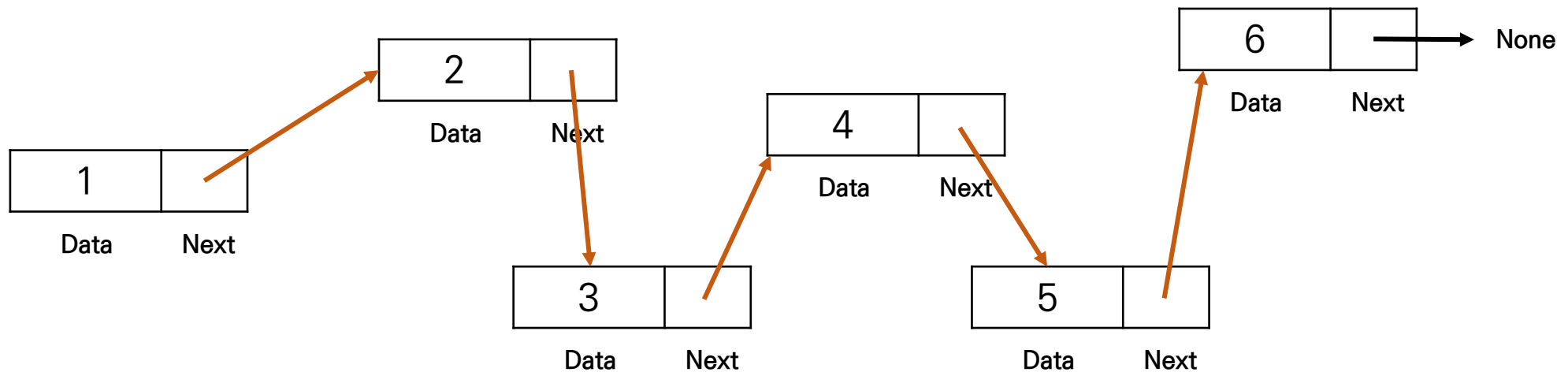
- 인덱스(Index)를 통해 데이터에 빠르게 접근
- 배열의 길이는 변경 불가능 → 길이를 변경하고 싶다면 새로 생성
- 데이터 타입은 고정

메모리 주소	1000	1004	1008	1012	1016	1020	1024
데이터	1	2	3	4	5	6	7
인덱스	A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]

2) 연결 리스트 (Linked List)

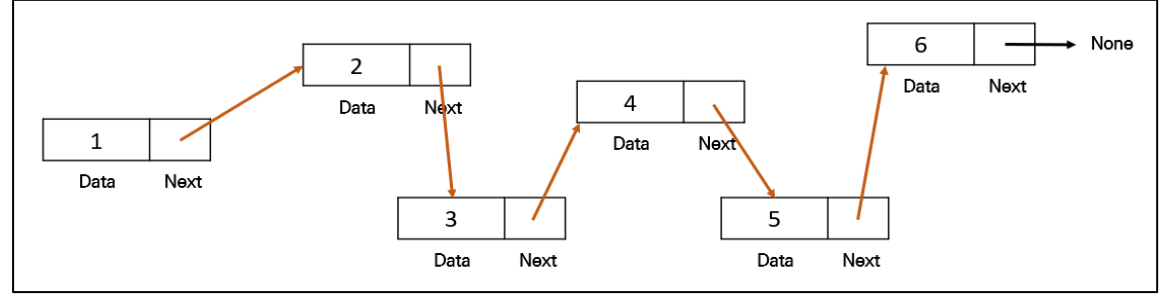
데이터가 담긴 여러 노드들이 순차적으로 연결된 형태의 자료구조

- 맨 처음 노드부터 순차적으로 탐색
- 연결리스트의 길이 자유롭게 변경 가능 → 삽입, 삭제가 편리
- 다양한 데이터 타입 저장
- 데이터가 메모리에 연속적으로 저장되지 않음



1. 배열 vs 연결리스트

메모리 주소	1000	1004	1008	1012	1016	1020	1024
데이터	1	2	3	4	5	6	7
인덱스	A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]



배열

연결리스트

인덱스 접근

가변 길이

파이썬의 리스트 (List)

인덱스	A[0]	A[1]	A[2]	A[3]	A[4]	A[5]
주소	2456	3882	6428	1003	2938	8472
데이터	0	1.5	2	3	"a"	[1, 2]



2. 파이썬의 리스트

파이썬 리스트의 메서드

- 1) `.append()`
- 2) `.pop()`
- 3) `.count()`
- 4) `.index()`
- 5) `.sort()`
- 6) `.reverse()`

1) .append(원소)

리스트 맨 끝에 새로운 원소 **삽입**

```
a = [1, 2, 3, 4, 5]
a.append(6)
print(a)
```

[1, 2, 3, 4, 5, 6]

```
a = [1, 2, 3, 4, 5]
a.append(["a", "b"])
print(a)
```

[1, 2, 3, 4, 5, ['a', 'b']]

2) .pop(인덱스)

특정 인덱스에 있는 원소를 **삭제 및 반환**

```
a = [1, 2, 3, 4, 5]
b = a.pop()
print(a)
print(b)
```

[1, 2, 3, 4]
5

```
a = [1, 2, 3, 4, 5]
b = a.pop(2)
print(a)
print(b)
```

[1, 2, 4, 5]
3

3) .count(원소)

리스트에서 해당 원소의 **개수**를 반환

```
a = [1, 2, 2, 3, 3, 3]
print(a.count(2))
```

2

```
a = [1, 2, 2, 3, 3, 3]
print(a.count(3))
```

3

4) .index(원소)

리스트에서 처음으로 원소가 등장하는 **인덱스** 반환

```
a = [1, 2, 3, 2, 5]
print(a.index(2))
```

1

```
a = [1, 2, 3, 2, 5]
print(a.index(8))
```

ValueError

Input **In [5]**, in <cell 1
1 a = [1, 2, 3, 2,
----> 3 print(a.index(8))

ValueError: 8 is not in

5) .sort()

리스트를 오름차순으로 정렬

`reverse=True` 옵션을 통해 내림차순으로 정렬 가능

```
a = [5, 2, 4, 0, -1]
a.sort()
print(a)
```

`[-1, 0, 2, 4, 5]`

```
a = [5, 2, 4, 0, -1]
a.sort(reverse=True)
print(a)
```

`[5, 4, 2, 0, -1]`

6) .reverse()

리스트의 원소들의 순서를 거꾸로 뒤집기

```
a = [1, 2, 3, 4, 5]
a.reverse()
print(a)
```

`[5, 4, 3, 2, 1]`

자주 쓰이는 리스트 관련 내장함수

- 1) `len()`
- 2) `sum()`
- 3) `max()`
- 4) `min()`
- 5) `sorted()`
- 6) `reversed()`

1) len(*iterable*)

리스트의 **길이**(원소의 개수)를 반환

```
a = [1, 2, 3, 4, 5]  
print(len(a))
```

5

2) sum(*iterable*)

리스트의 모든 원소의 **합**을 반환

```
a = [1, 2, 3, 4, 5]  
print(sum(a))
```

15

3) `max(iterable)`

리스트의 원소 중 **최대값**을 반환

```
a = [1, 2, 3, 4, 5]
print(max(a))
```

5

4) `min(iterable)`

리스트의 원소 중 **최소값**을 반환

```
a = [1, 2, 3, 4, 5]
print(min(a))
```

1

5) sorted(*iterable*)

오름차순으로 **정렬**된 새로운 리스트 반환
원본 리스트는 변화 없음

```
a = [5, 2, -1, 0, 1]
b = sorted(a)
c = sorted(a, reverse=True)
```

```
print(a) # 원본
print(b) # 오름차순 정렬
print(c) # 내림차순 정렬
```

```
[5, 2, -1, 0, 1]
[-1, 0, 1, 2, 5]
[5, 2, 1, 0, -1]
```

6) reversed(*iterable*)

리스트의 순서를 거꾸로 **뒤집은** 새로운 객체 반환
원본 리스트는 변화 없음

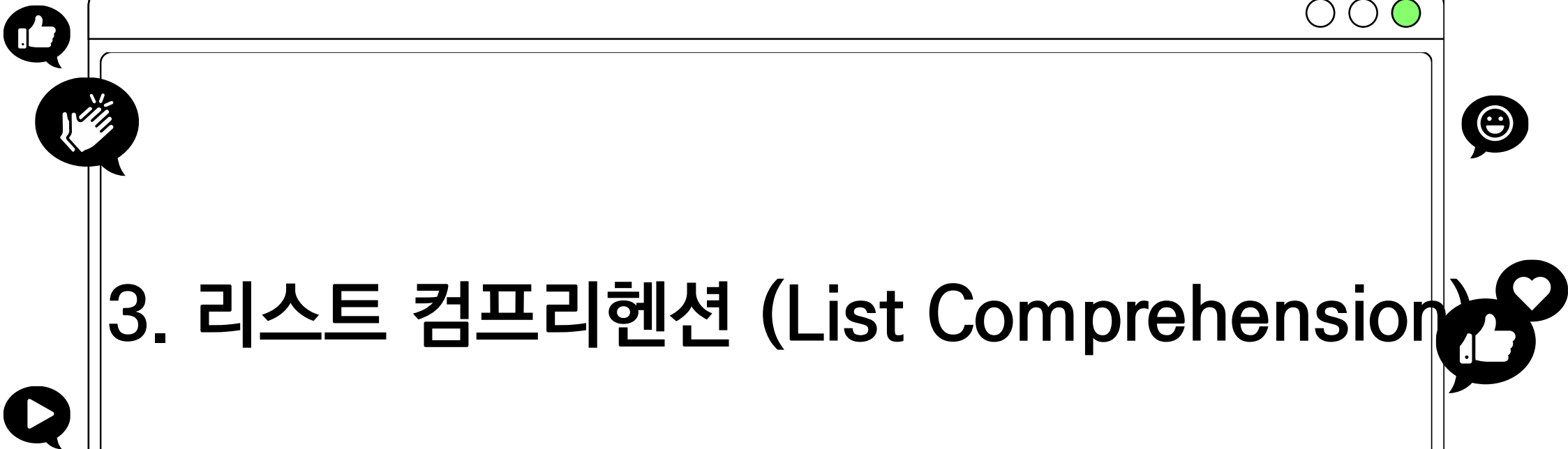
```
a = [1, 2, 3, 4, 5]
b = reversed(a)
c = list(reversed(a))
```

```
print(a) # 원본
print(b) # reversed(a)
print(c) # list(reversed(a))
```

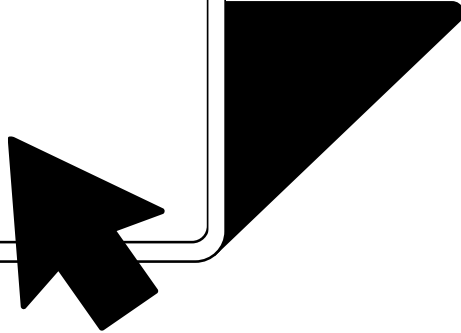
```
[1, 2, 3, 4, 5]
<list_reverseiterator object at 0x000002980CE25E740>
[5, 4, 3, 2, 1]
```

리스트 관련 내장함수 연습

문제 번호	문제	링크
BOJ 10818	최소, 최대	https://www.acmicpc.net/problem/10818
BOJ 11720	숫자의 합	https://www.acmicpc.net/problem/11720
BOJ 2750	수 정렬하기	https://www.acmicpc.net/problem/2750



3. 리스트 컴프리헨션 (List Comprehension)



3. 리스트 컴프리헨션 (List Comprehension)

List Comprehension(리스트 컴프리헨션, 리스트 내포)란
코드 **한 줄**만으로 새로운 리스트를 만드는 방법이다.

```
numbers = []  
for i in range(5):  
    numbers.append(i)
```

```
print(numbers)
```

[0, 1, 2, 3, 4]



```
numbers = [i for i in range(5)]
```

```
print(numbers)
```

[0, 1, 2, 3, 4]

```
odd_numbers = [i for i in range(10) if i % 2 == 1]
```

```
print(odd_numbers)
```

[1, 3, 5, 7, 9]

if문으로 필터링도 가능

3. 리스트 컴프리헨션 (List Comprehension)

리스트 컴프리헨션 연습 - 1

문제

[목록](#)

정수 n 을 입력받아, 1부터 n 까지의 제곱수를 리스트로 만들어 출력하는 프로그램을 작성하라.
리스트 내포를 이용한다.

입력 예

5

출력 예

[1, 4, 9, 16, 25]

3. 리스트 컴프리헨션 (List Comprehension)

리스트 컴프리헨션 연습 - 2

문제

리스트 내포를 이용하여 1번부터 입력받은 숫자까지의 숫자에 "No."를 붙인 문자열을 원소로 하는 리스트를 만들어 출력하시오.

입력 예

5

출력 예

`['No.1', 'No.2', 'No.3', 'No.4', 'No.5']`

http://www.jungol.co.kr/bbs/board.php?bo_table=pbank&wr_id=4356&sca=pyc0

3. 리스트 컴프리헨션 (List Comprehension)

리스트 컴프리헨션 연습 - 3

문제 번호	문제	링크
BOJ 2562	최댓값	https://www.acmicpc.net/problem/2562