

[자료구조 1. Arrays]

[Array]

- 메모리에서의 연속적인 공간
- 같은 데이터 타입의 모음
- 인덱스와 그에 매칭되는 값의 쌍 묶음

[Array in Python]

- python에서는 List라고도 한다.
- 리스트의 생성: `list_name = []`
- 리스트의 인덱싱:

`list_name[0]` (0번째 원소 접근)

`list_name[-1]` (마지막 원소 접근)

- 리스트의 슬라이싱:

`List_name[start index, end index, interval]`

마지막 인덱싱은 포함되지 않는다 (ex, `1:3` = 1, 2 인덱스만 표현)

- 리스트의 연산:

리스트 + 리스트: 리스트의 값을 이어줌

리스트 * 정수: 리스트를 정수 번 반복

리스트의 연산은 다른 타입의 원소끼리는 불가능

- 자주 쓰는 코드

1. 리스트 길이: `len(list_name)`

2. 리스트 원소 수정: `list_name[0] = val`

3. 리스트 원소 추가1: `list_name.append(val)`

4. 리스트 원소 추가2: `list_name.insert(index, val)`

5. 리스트 원소 인덱스 삭제: `del list_name[index]`

6. 리스트 원소 전체 삭제: `list_name.clear()`

7. 리스트 원소 값 삭제: `list_name.remove(val)`

8. 리스트 원소 값 정렬1: `list_name.sort(reverse = True)` - 변경 적용

9. 리스트 원소 값 정렬2: `sorted(list_name)` - 정렬 출력, 변경 x

10. 리스트 뒤집기1: `list_name.reverse()`

11. 리스트 뒤집기2: `list(reversed(list_name))` - `list()`를 사용하지 않으면

list_reverseiterator 객체를 반환

12. 리스트 원소 꺼내기: list_name.pop() - FIFO 구조

13. 리스트 원소 수 세기: list_name.count()

14. 리스트 복사하기: list.copy() - 깊은 복사(복사된 값을 변경해도 원본 변경 x)