

[Daily Python GRAMMAR: Linked List]

연결리스트: 다른 추상 자료형을 구현할 때 기반이 되는 기초 선형 자료구조

각 노드가 데이터 포인터를 가지고 한 줄로 연결되는 방식(다음 노드를 가리키는 포인터)

[배열 vs 연결리스트]

- 배열은 물리적인 메모리 주소가 연속적이고 연결리스트는 물리 메모리 주소가 연속적이지 않고 임의이다.
- 배열의 삽입/삭제: $O(n)$ - 해당 위치 이후의 모든 원소들을 이동시키는데 시간이 소요
- 연결리스트의 삽입/삭제: $O(1)$ - 해당 노드 이외의 다른 노드들에 영향 x(가리키는 주소만 변경)
- 배열의 접근: $O(1)$ - 각 원소의 인덱스 이용
- 연결리스트의 접근: $O(n)$ - 연결된 메모리가 아니므로 모든 노드를 거쳐서 탐색

[연결리스트 정의]

```
class ListNode:
    def __init__(self, val=0, next=None):
        self.val = val
        self.next = next
```

[연결리스트 생성]

- 헤더를 선언해 연결리스트 생성, 헤더를 통해 다른 모든 노드를 탐색 및 참조
- 헤드를 이동시키지 않고 node=head로 head 주소를 참조하여 사용

```
head = ListNode(0)
```

[노드 탐색 및 삭제]

```
node=head
while node.next:
    if node.next.val==2:
        next_node=node.next.next
        node.next=next_node
        break
    node=node.next
```

```
node=head
while node:
    print(node.val)
    node=node.next
```