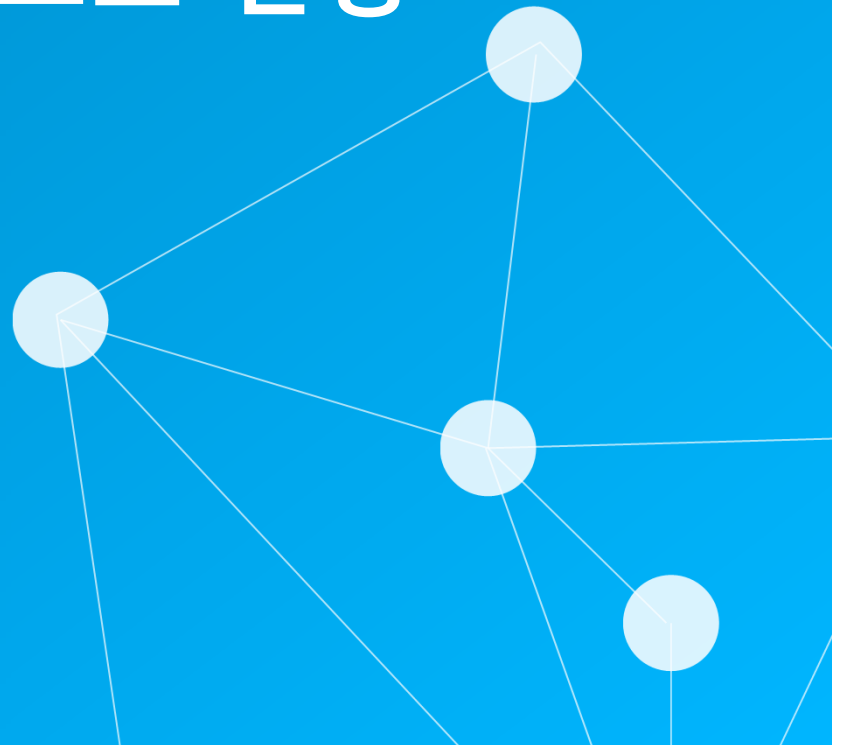


연결리스트 변형



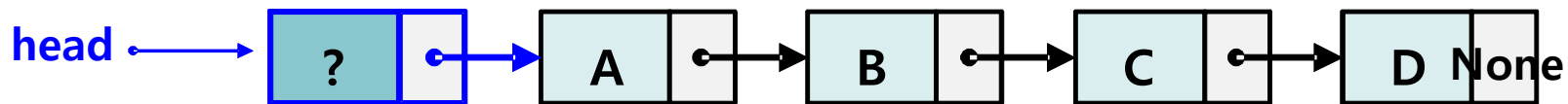
연결리스트 변형들

- 단순연결리스트 (singly linked list)
- 원형 연결리스트 (circular linked list)
- Header 노드가 있는 연결리스트
- 이중연결리스트 (doubly linked list)

헤더 노드가 있는 연결리스트



헤더노드(dummy node)



연결리스트의 처음에 더미(dummy) 노드를 둬

- 더미노드에는 연결리스트 정보(예를 들어 길이)를 저장

- 연결리스트의 처음에 노드를 삽입할 경우나 첫번째 노드를 삭제할 경우에 대한 특별한 처리를 하지 않아도 된다.

만약 삭제할 경우 중간에 삭제할 경우

나를 필요

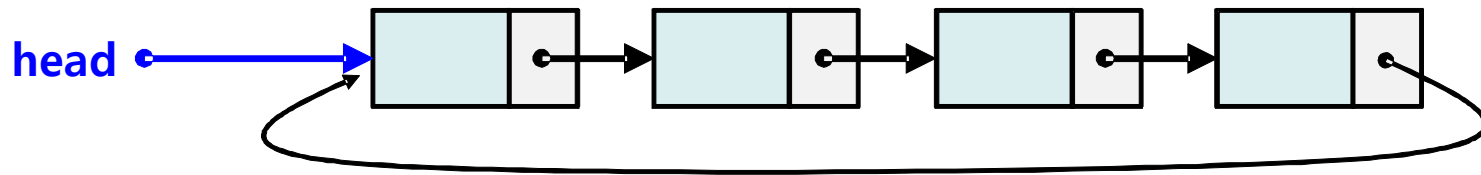
=> 코드가 간단해짐.

- 연결리스트에 노드를 삽입하거나 삭제할 때, 때, 리스트가 empty 인 경우를 특별하게 처리하지 않아도 된다.

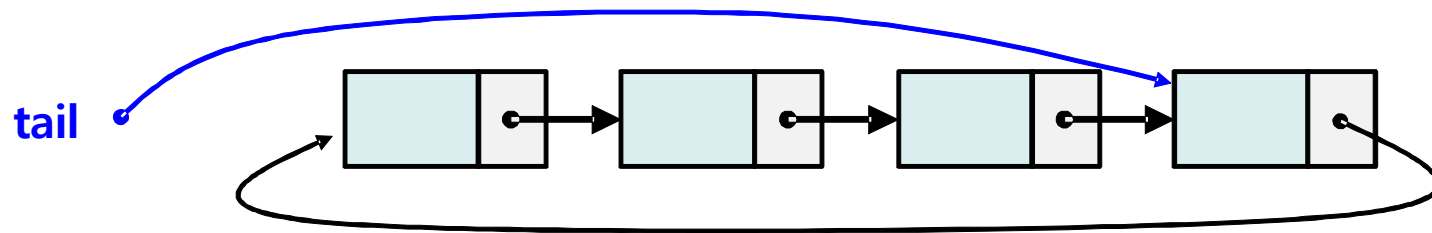
- 연결리스트 삽입/삭제처리를 간단하게 하는데 사용된다.

원형 연결 리스트 (circular linked list)

- Circular Linked List

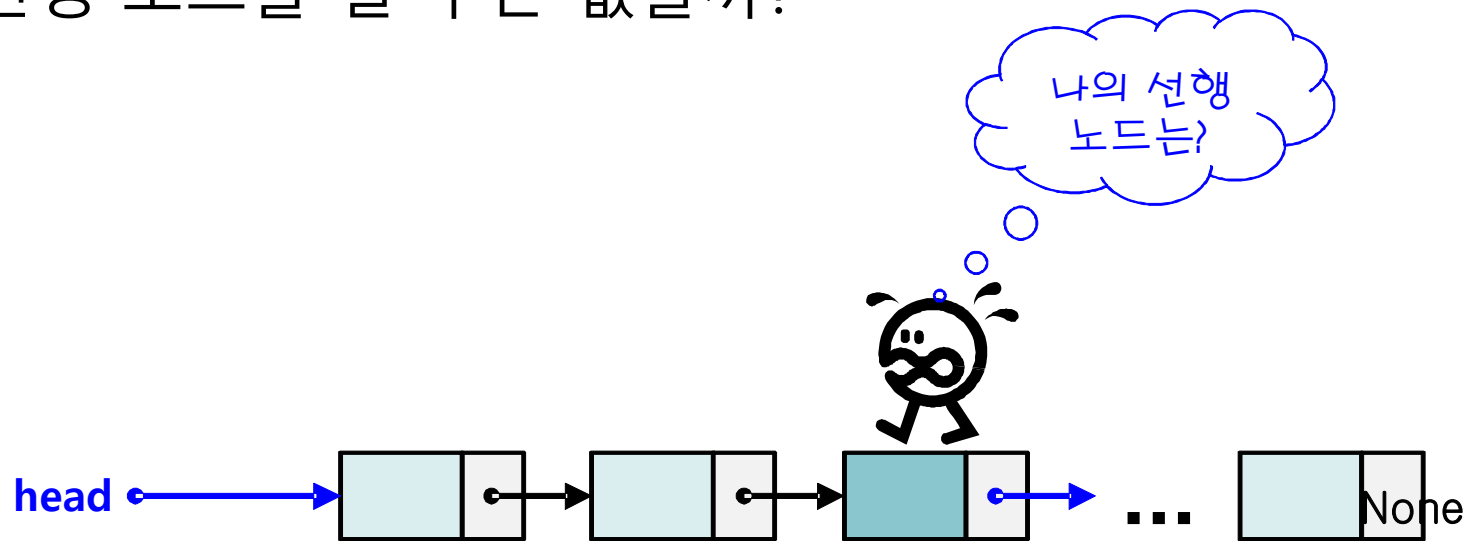


- Circular Linked List



이중 연결 리스트 (doubly linked list)

- 후속 노드는 쉽게 알 수 있다.(링크 필드)
- 선행 노드를 알 수는 없을까?



이중연결리스트

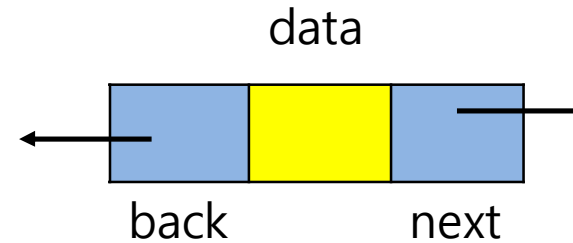
. 각 노드는 다음 노드를 가리키는 링크 필드와 이전 노드를 가리키는 링크 필드가 있다.



이중연결리스트 구현

```
class DNode: # 이중연결리스트 노드
    def __init__(self, e):
        self.addr = e
        self.next = None
        self.back = None
```

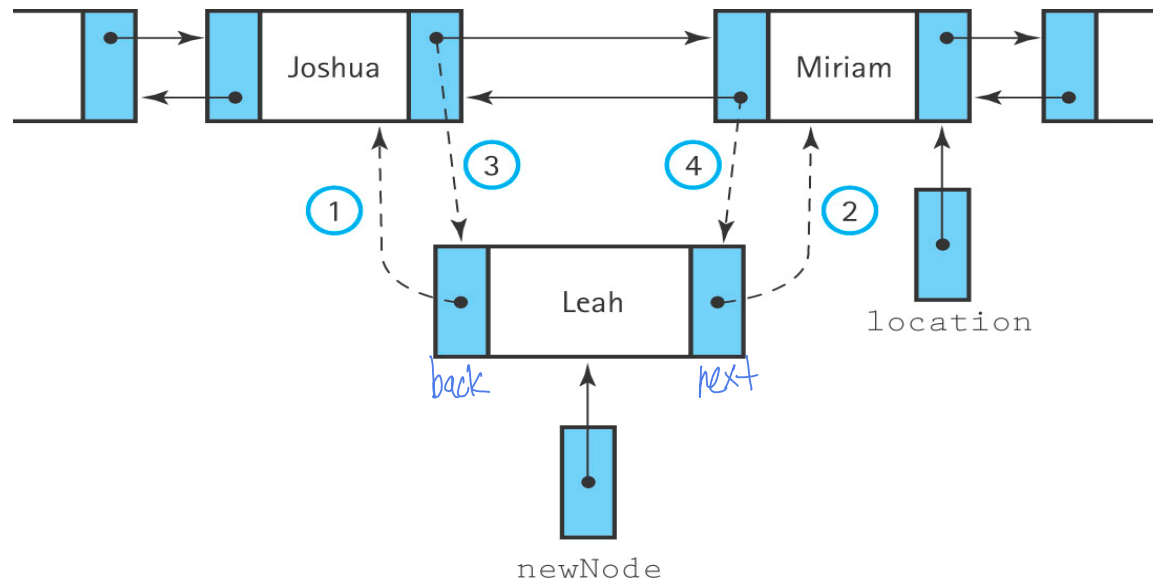
```
class DLinkedList: # 이중연결리스트
    def __init__(self):
        head = None
```



새로운 노드 삽입

. 연결 리스트에서 특정한 (location이 가리키는) 노드 바로 앞에 새로운 (newNode가 가리키는) 노드를 삽입하는 것이 간편하다.

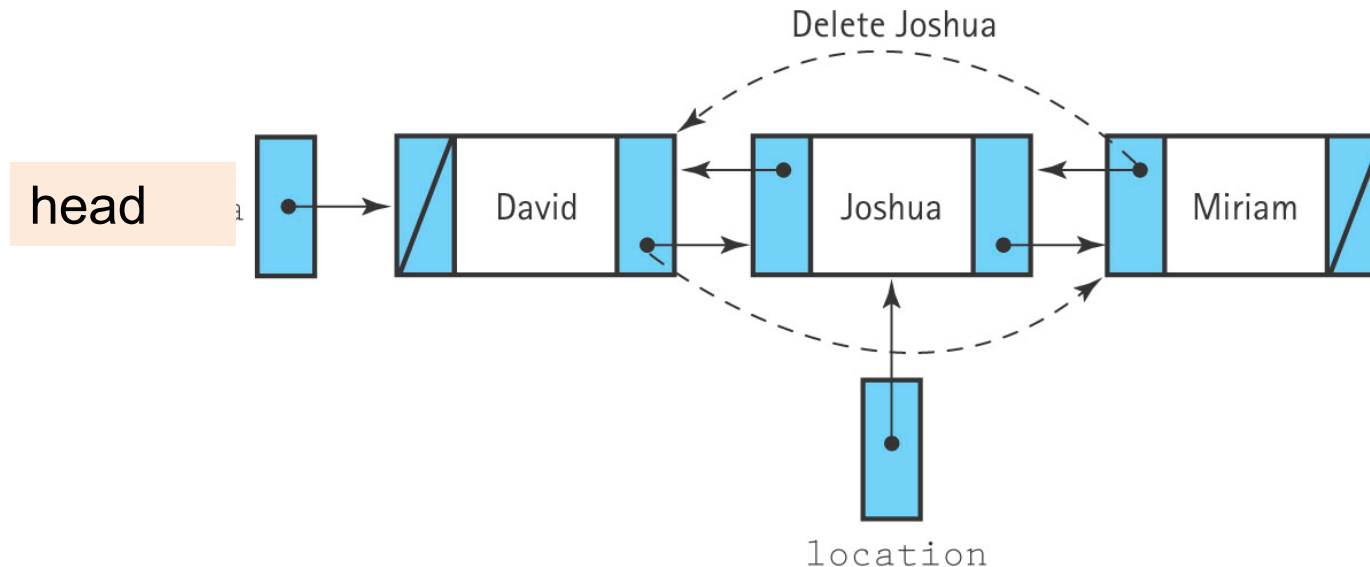
```
newNode.back = location.back ①  
newNode.next = location      ②  
if (location.back == None): # 처음에 노드 삽입  
    head = newNode  
else:  
    location.back.next = newNode ③  
  
location.back = newNode ④
```



노드 삭제

- 특정한 (location이 가리키는) 노드를 삭제하는 것이 간편함

```
if (location.back == None): # 첫번째 노드 삭제
    head = location.next;
    if (location.next != None): # 첫번째 노드가 마지막 노드가 아닌 경우
        location.next.back = None
else: # 첫번째 노드 삭제가 아닌 경우
    location.back.next = location.next
    if (location.next != None): # 마지막 노드 삭제가 아닌 경우 (중간 노드 삭제)
        location.next.back = location.back
```



이중연결리스트 장/단점

- 장점

- ✓ 특정한 노드 이전에 새로운 노드를 삽입하는 것이 간편하다.
- ✓ 특정한 노드를 삭제하는 것이 간편하다.
- ✓ 현재 노드에서 이전 노드로 이동하는 것이 간편하다.

- 단점

- ✓ 메모리 공간: Singly linked list의 두 배

Linked list 여러 변형들

- circular linked list
- 헤더 노드가 있는 (singly) linked list
- 헤더 노드가 있는 circular linked list
- circularly doubly linked list
- 헤더 노드가 있는 doubly linked list
- 헤더 노드가 있는 circularly doubly linked list