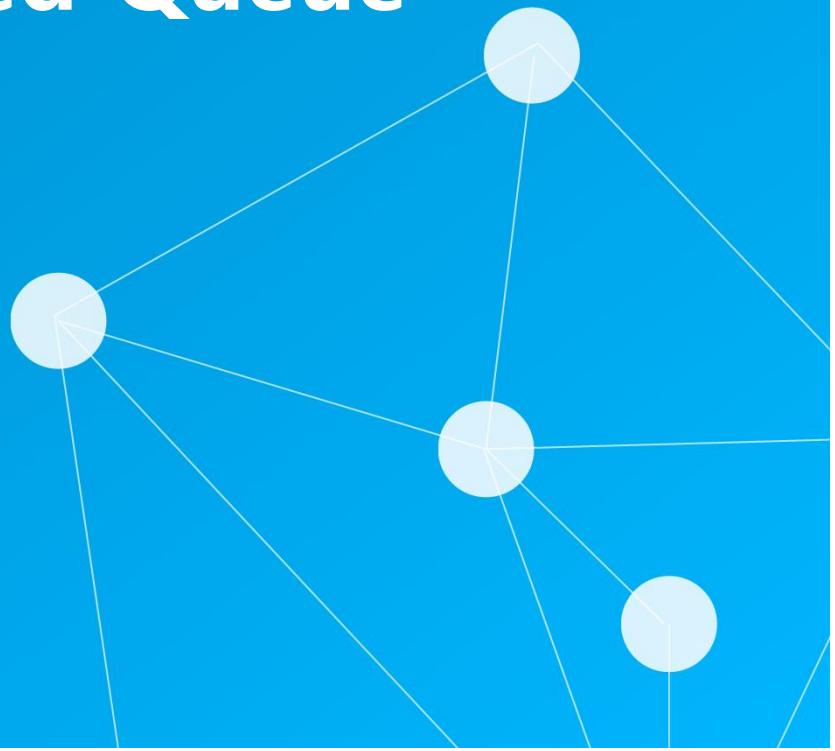
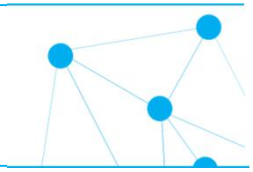
파이썬
자료구조

CHAPTER

Linked Queue

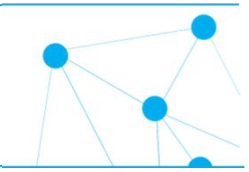


6.1 연결구조란?

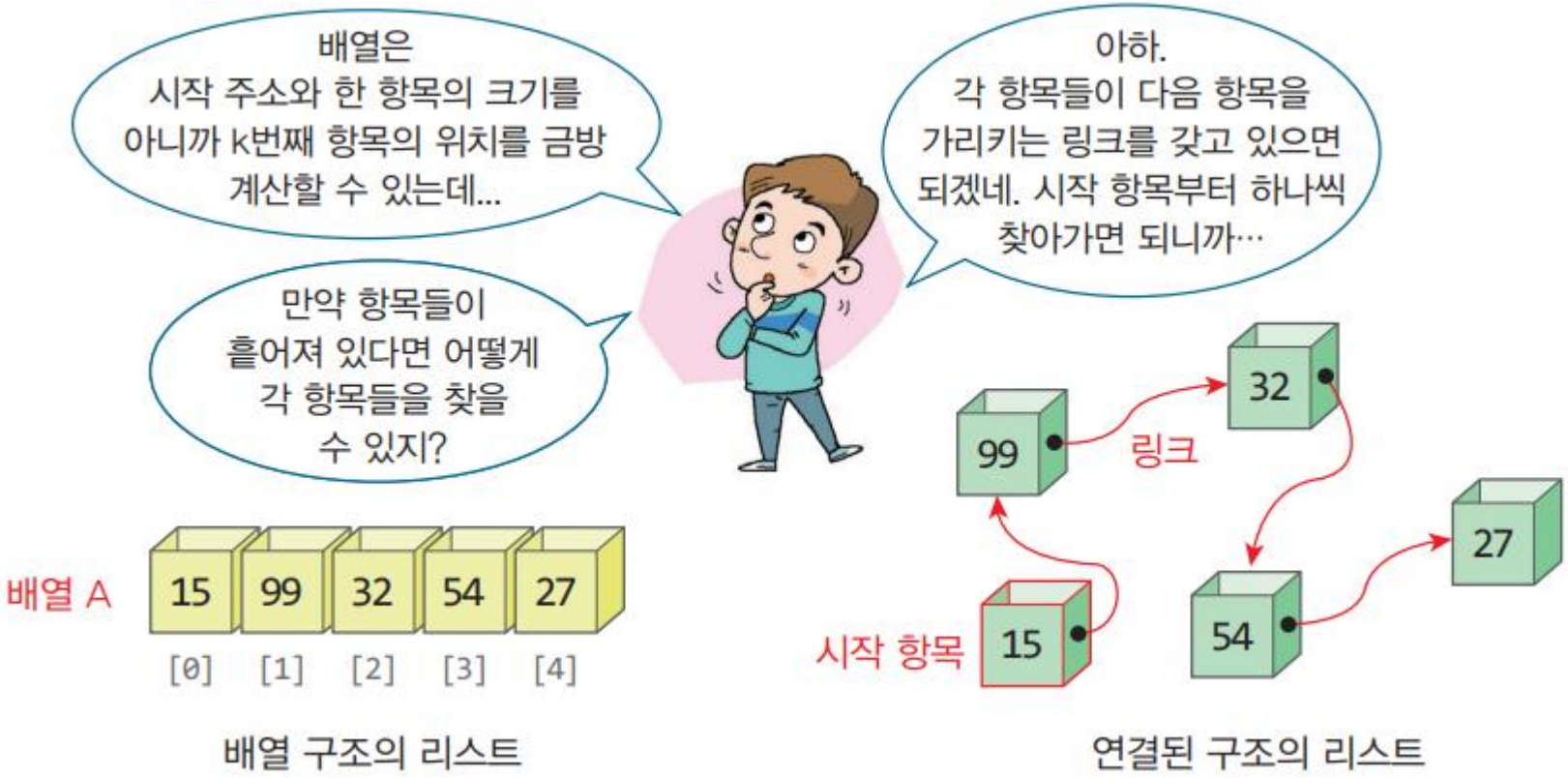


- 연결구조는 흩어진 데이터를 링크로 연결해서 관리한다.
- 연결구조의 특징
- 연결리스트의 구조
- 연결리스트의 종류

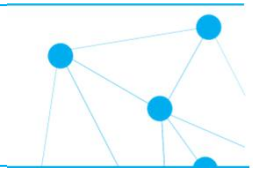
연결된 구조란?



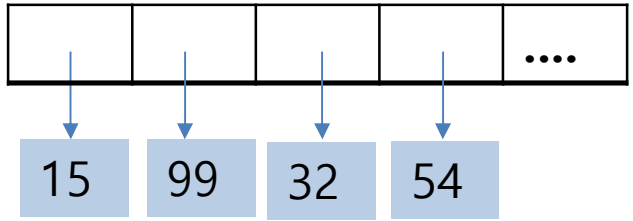
- 연결된 구조는 흩어진 데이터를 링크로 연결해서 관리



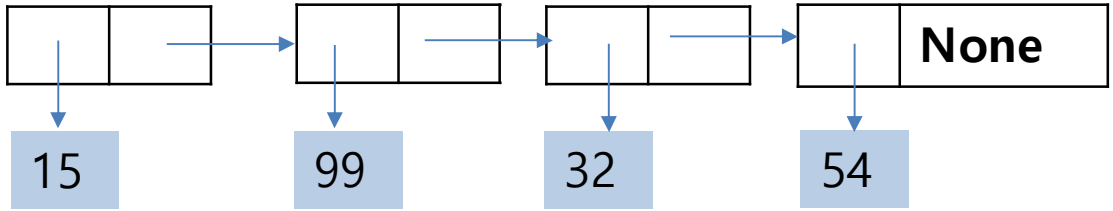
연결구조의 예



파이썬 리스트 (배열)

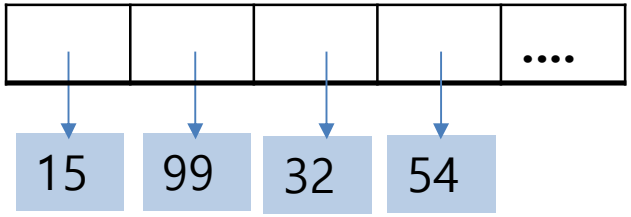


파이썬 연결구조(연결리스트)



연결구조의 예

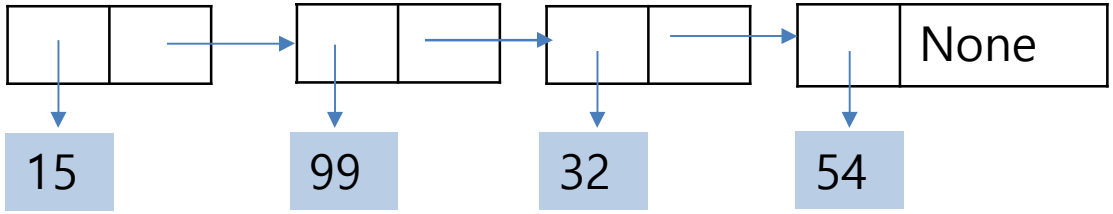
파이썬 리스트 (배열)



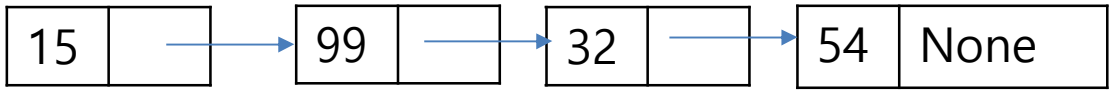
편의상 아래와 같이 나타낸다

15	99	32	54	....

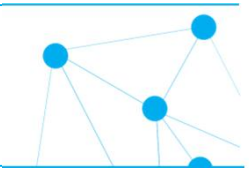
파이썬 연결리스트



위의 연결리스트를 편의상 아래와 같이 나타낸다



6.3 Linked Queue



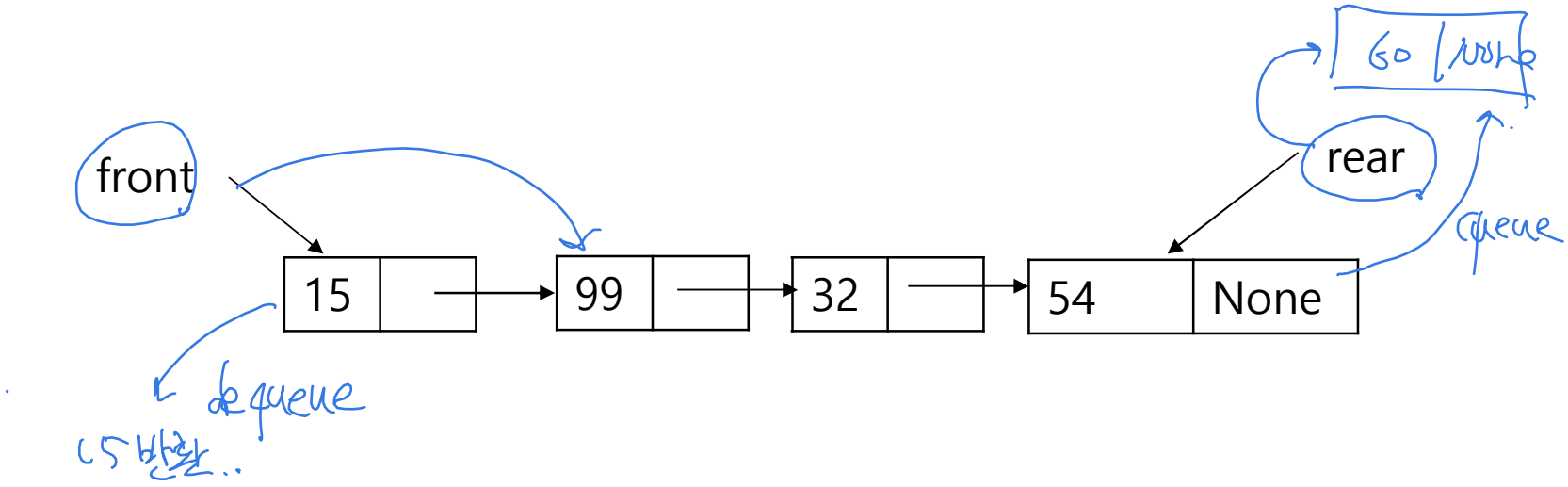
- 노드

- 노드 클래스

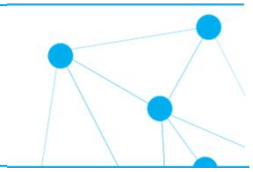
```
class Node:
    def __init__(self,element):
        self.data = element
        self.link = None
```

Linked 큐

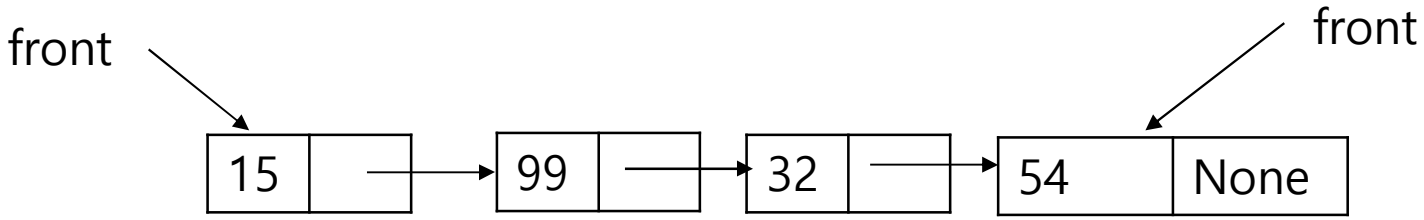
- 연결된 큐 클래스
- 앞 (15, 99, 32, 54) 뒤



Linked 큐

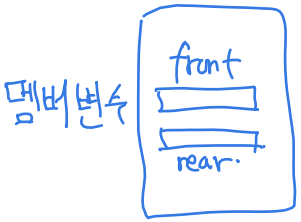


- 연결된 큐 클래스
- 앞 (15, 99, 32, 54) 뒤

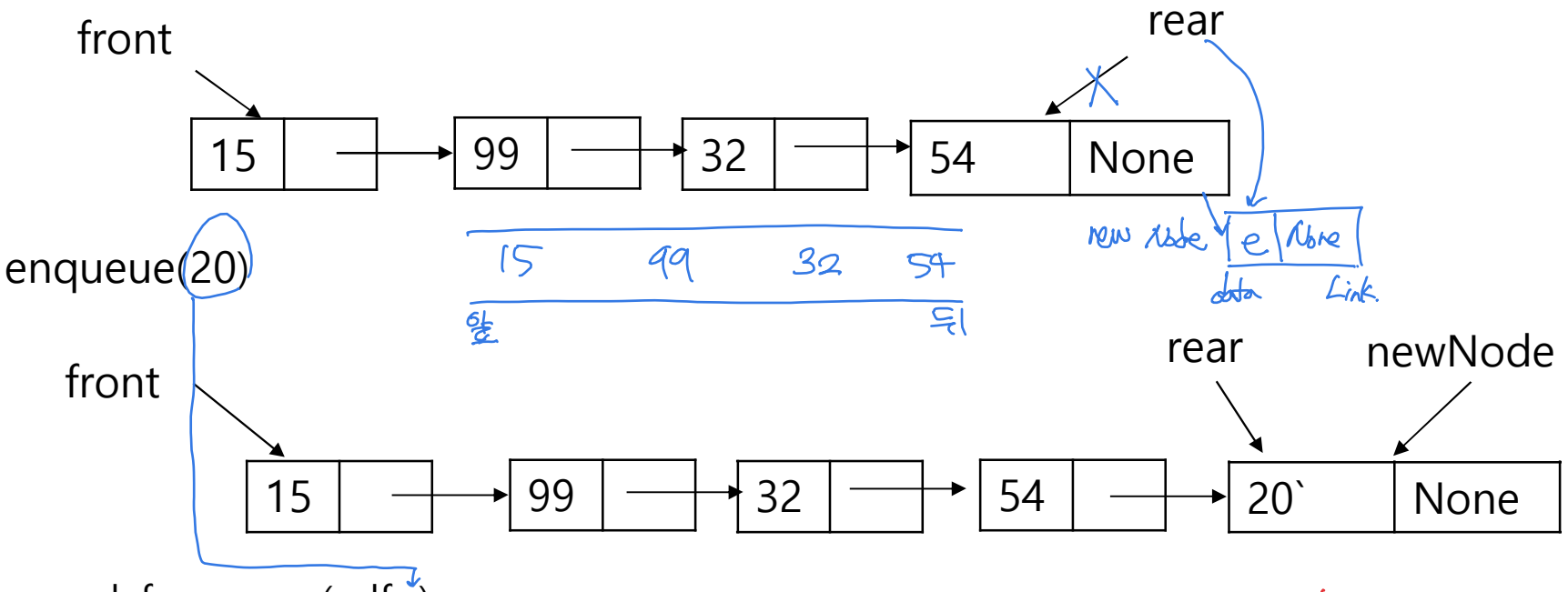



```
class LinkedQueue:
    def __init__(self):
        self.front = self.rear = None


    def isEmpty(self):
        return self.front == None
```

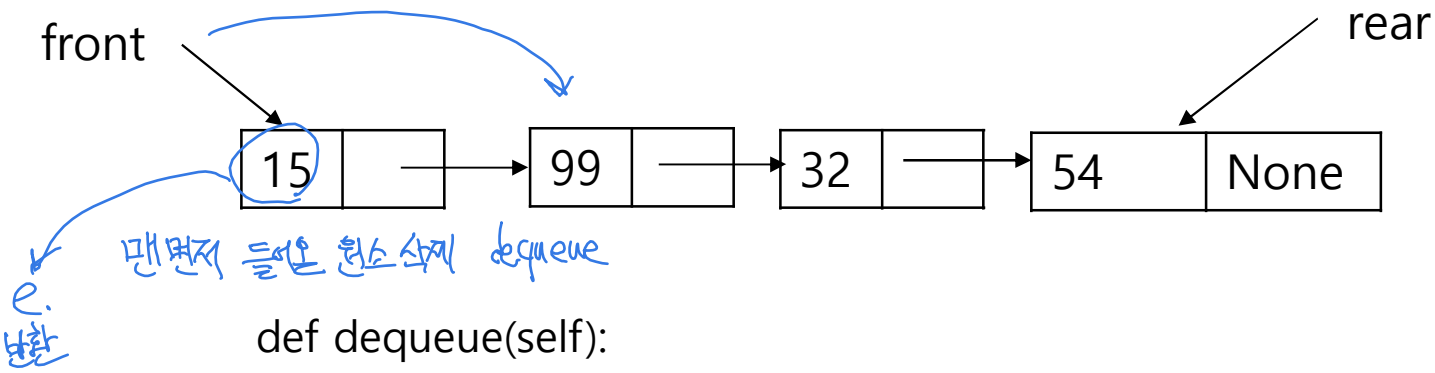


삽입 연산 enqueue




```
def enqueue(self,e):
    newNode = Node(e)
    if self.front == None:
        self.front = self.rear = newNode
        return
    else:
        self.rear.link = newNode
        self.rear = newNode
```

삭제 연산 dequeue



맨 먼저 들어온 원소 삭제 dequeue

e. 반환

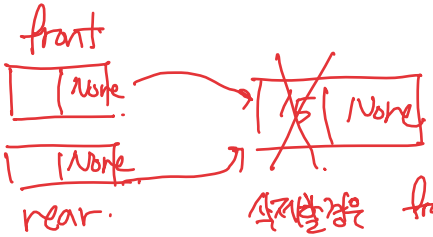
```
def dequeue(self):

    if self.isEmpty():
        print("Queue empty")
        return

    e = self.front.data
    self.front = self.front.link
    if self.front == None:
        self.rear = None

    return e
```

원소가 하나만 남아있는 경우



삭제할 경우 front rear 모두 None으로 만들어줌.

예외적인 경우 별도 추가