



데이터 구조

4주차 : 이중연결리스트 코딩

이중연결리스트 필요 메소드

1. HEAD 생성
2. 마지막 위치에 데이터 추가
3. 원하는 위치에 데이터 추가
4. 데이터 전체 순차/역순 출력
5. 데이터 삭제
6. 데이터 수정
7. 원하는 위치 검색

이중연결리스트

■ Node 클래스

```
class Node:  
    def __init__(self, data=None):  
        self.lnext = None  
        self.data = data  
        self.rnext = None
```

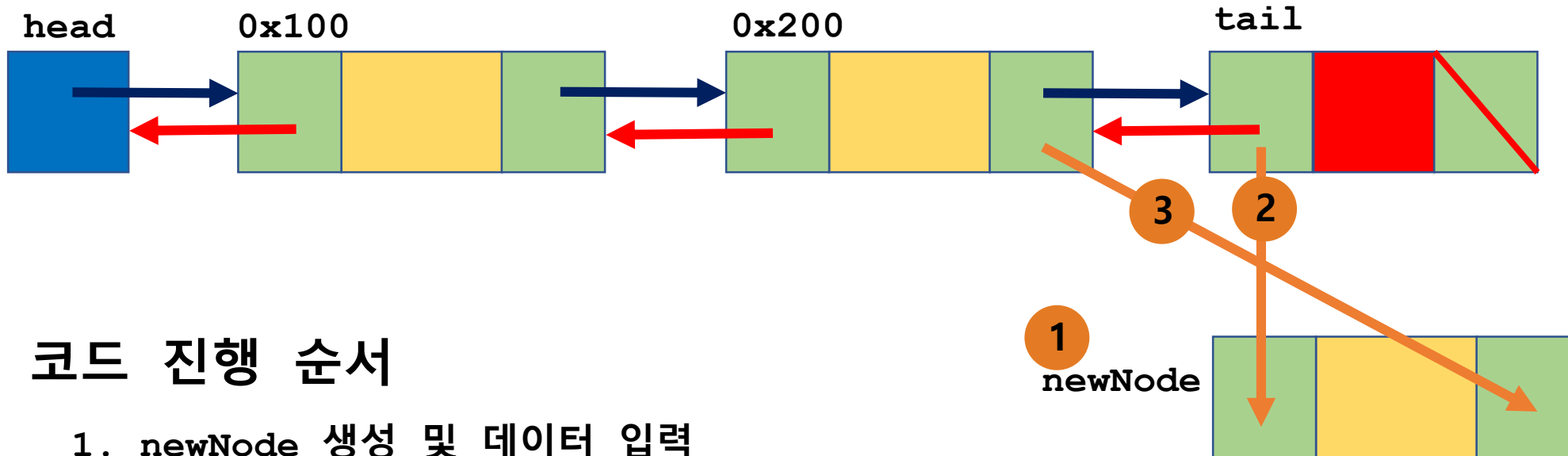
■ Head / Tail 생성 코드

```
def create_ht(self):  
    self.head = self.Node()  
    self.tail = self.Node()
```

Node 구조



▣ 마지막 위치에 데이터 추가 (첫 데이터 입력 시 제외)

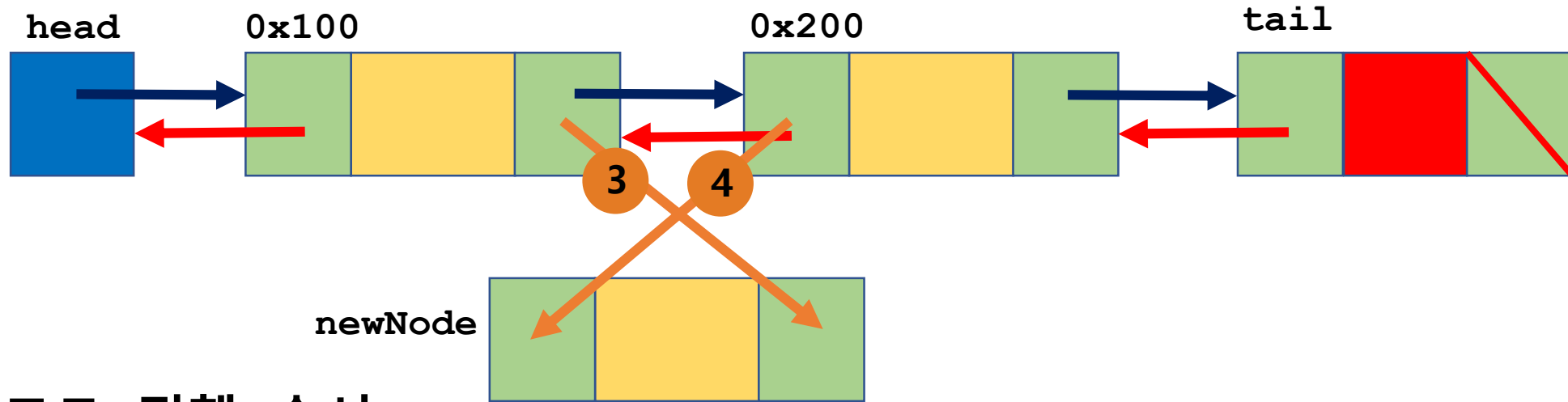


※ 코드 진행 순서

1. newNode 생성 및 데이터 입력
2. `newNode.lnext = self.tail.lnext`
3. `newNode.rnext = self.tail.lnext.rnext`
4. `self.tail.lnext.rnext = newNode`
5. `self.tail.lnext = newNode`

이중연결리스트 주요 코드

▣ 원하는 위치에 데이터 추가 (tail 이전 노드 제외)



※ 코드 진행 순서

1. head 부터 순차 검색을 통한 원하는 데이터가 존재하는 Node 찾기
2. newNode 생성 및 데이터 입력
3. `newNode.lnext = current`
4. `newNode.rnext = current.rnext`
4. `current.rnext.lnext = newNode`
5. `current.rnext = newNode`

단순연결리스트 주요 코드

▣ 데이터 전체 순차 / 역순 출력 구현

▣ 데이터 수정, 삭제, 원하는 위치 탐색 구현