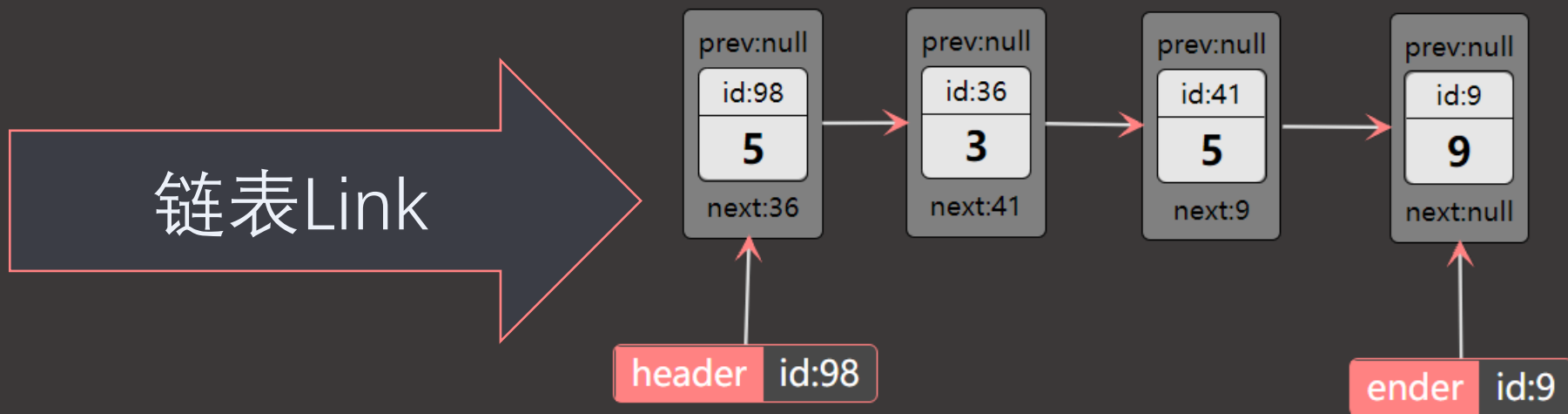


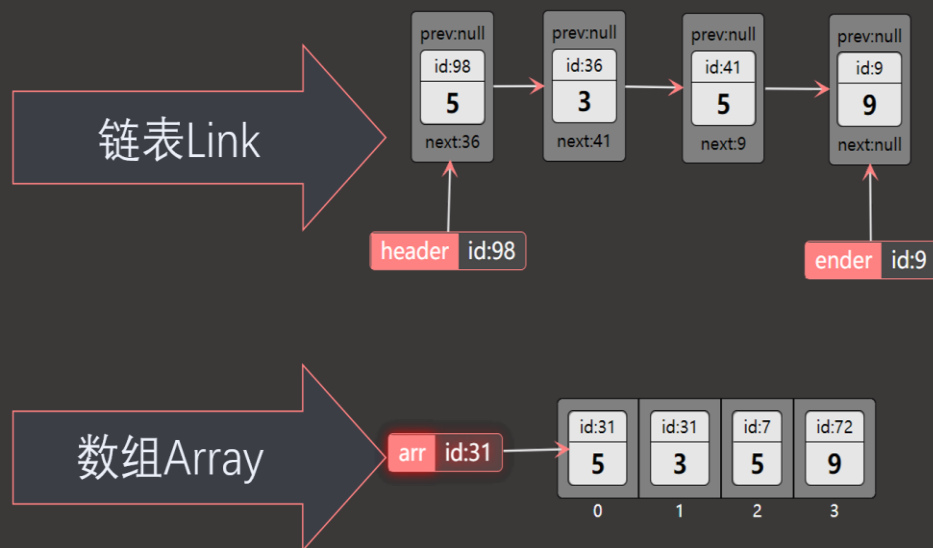
数据结构之

数组Array、链表Link、栈Stack、队列Queue

线性表List

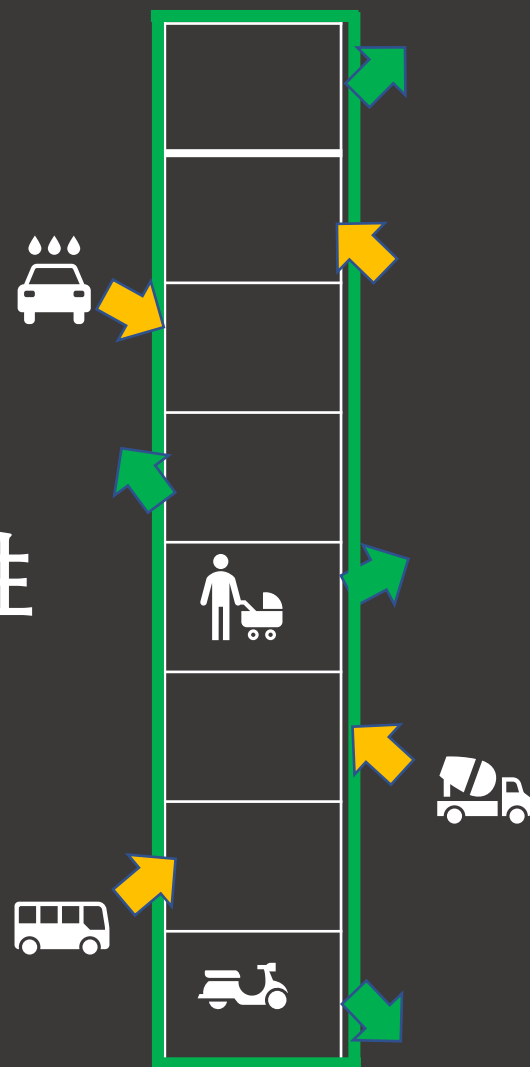


Array VS Link

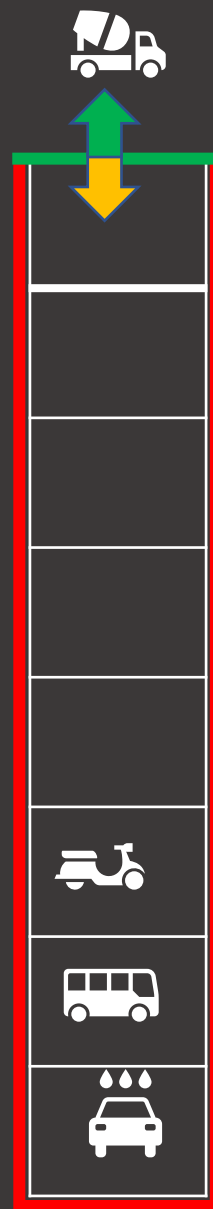


数组Array	链表Link
长度固定	长度可变
存储空间连续	存储空间不连续，靠指针连接
不适合频繁插入·删除	适合频繁插入·删除
不易实现栈与队列	易实现栈与队列

Lists' 的存储特性



链表Link
No Rule



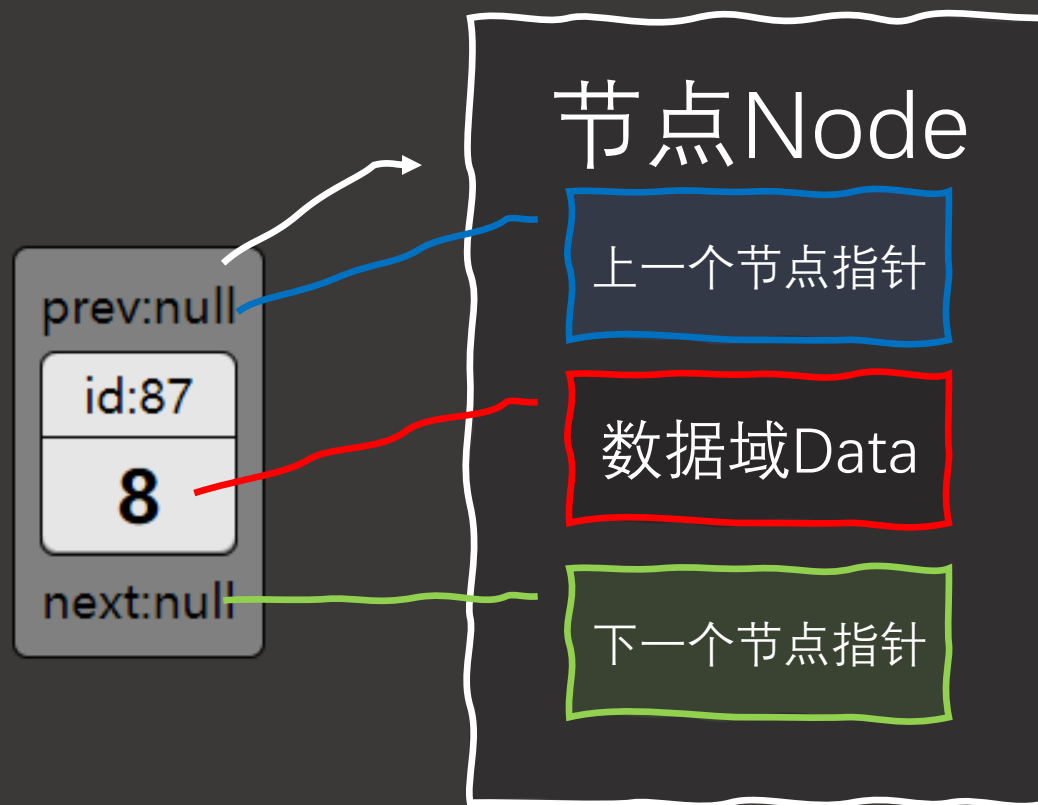
栈Stack
LIFO



队列Queue
FIFO

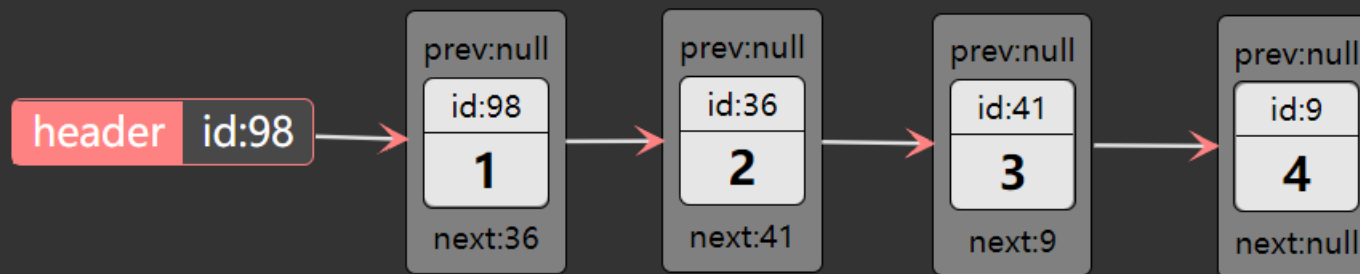
Link链表的相关概念

```
typedef struct node
{
    struct node * prev; // 上一个节点
    int data; // 数据域
    struct node * next; // 下一个节点
} Node, *P_NODE;
```



LinkedList的实现原理 **1**

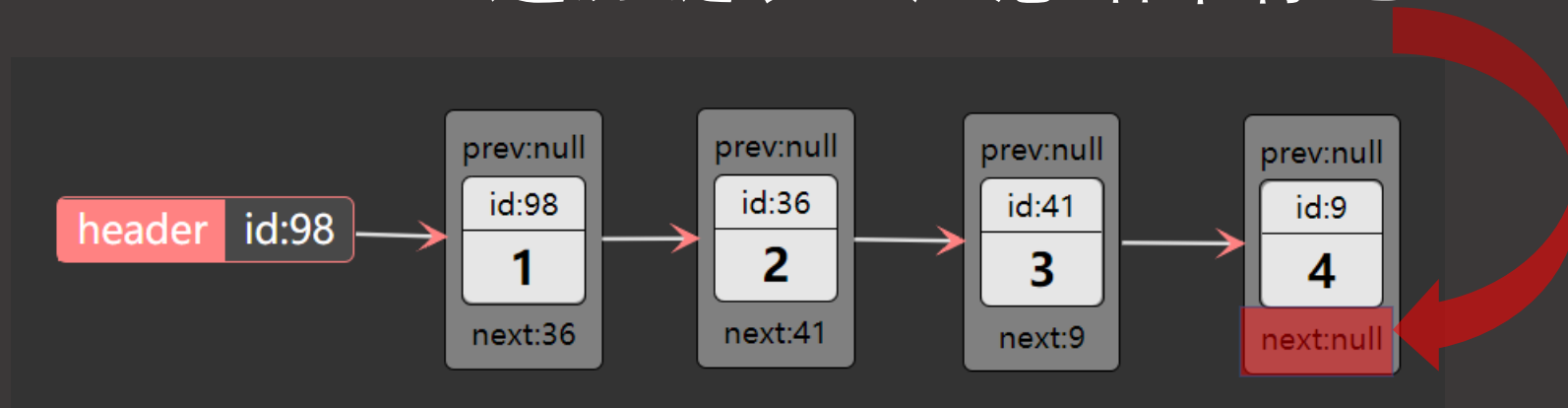
创建静态单向链表：理解节点连接



调试器观察内存情况

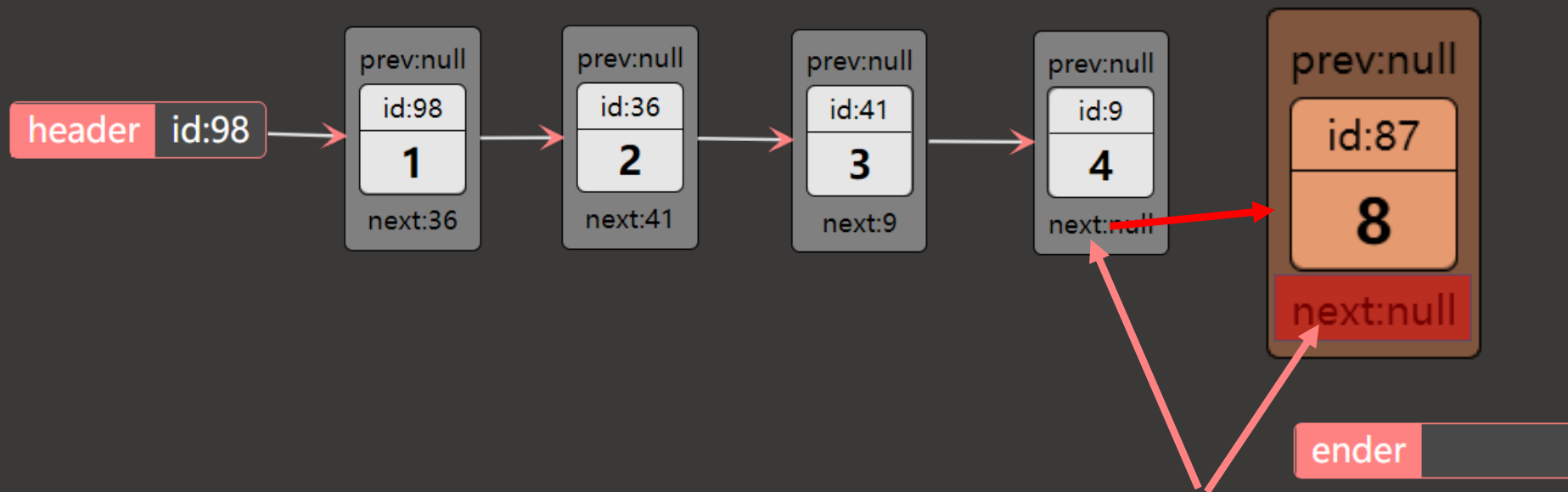
LinkedList的实现原理 **2**

遍历链表： 注意结束标记



LinkedList的实现原理 **3**

尾部追加： add

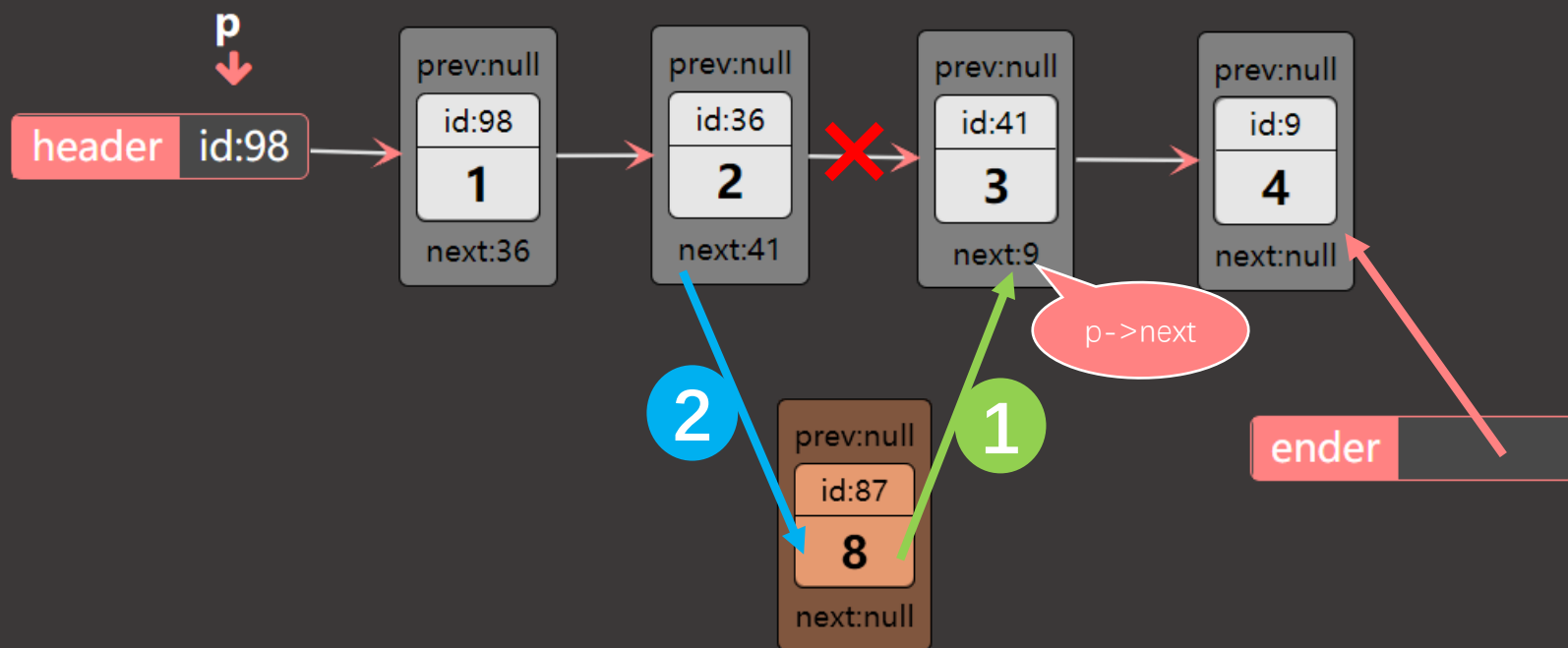


LinkedList的实现原理

4

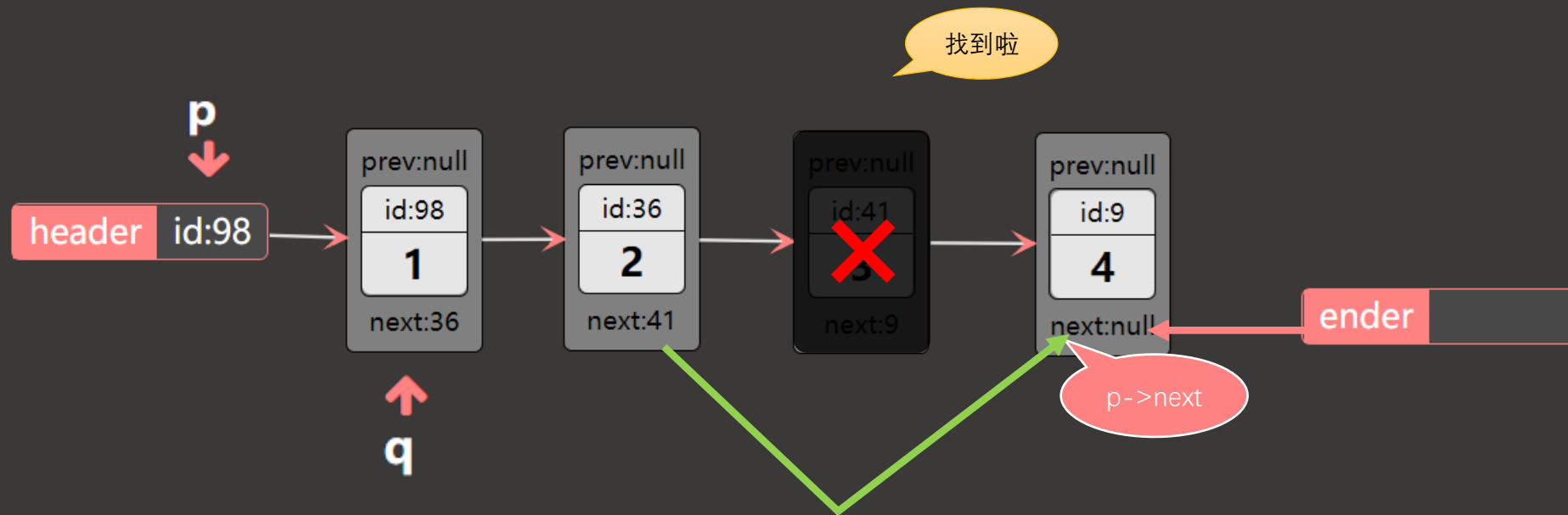
中间插入：insert

找到啦



LinkedList的实现原理 5

中间删除: remove



LinkedList的实现原理

6

查 找

size

节点个数

get

按下标找节点

indexOf

按内容找下标

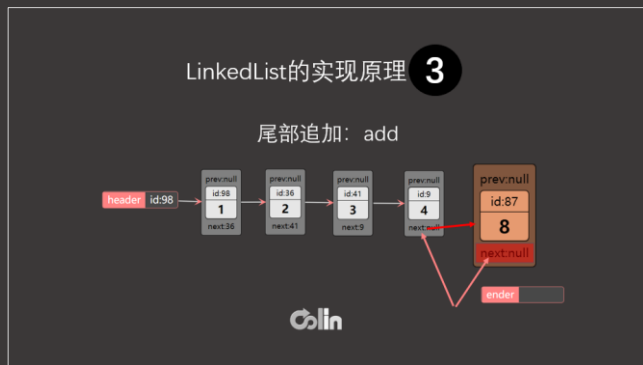
LinkedList的实现原理

2

遍历链表：注意结束标记



Link的高度封装



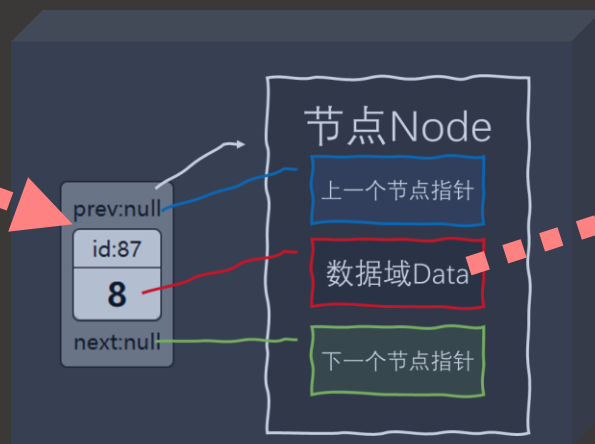
双向链表结构

节点结构

数据域结构

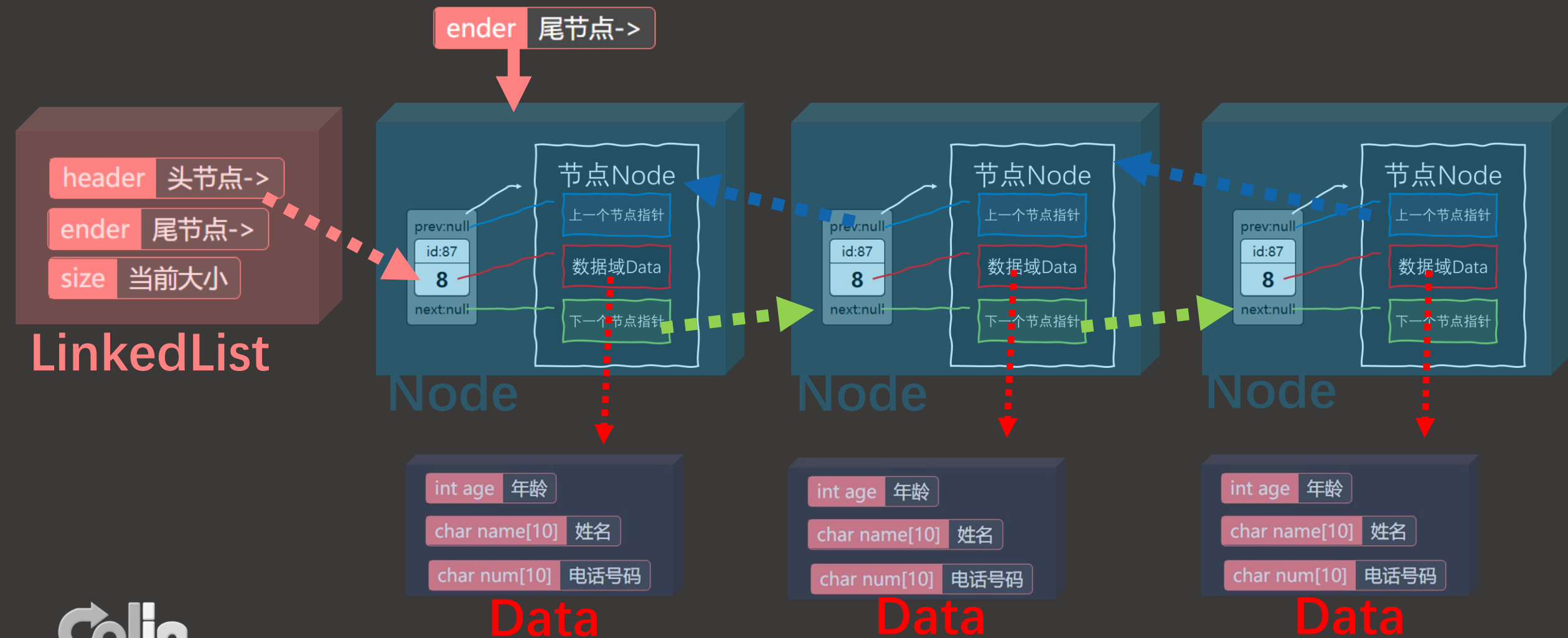


LinkedList



用户自行设计

LinkedList 数据存储形态



链表功能综合体

创建



追加
前插
后插

删除
清空

修改

Data找
下标找

Stack
实现

Queue
实现

返回处理结果

排序

交换

打乱

最大
最小

