1、什么是链表?

链表 [Linked List]：链表是由一组不必相连（不必相连：可以连续也可以不连续）的内存结构（节点），按特定的顺序链接在一起的抽象数据类型。

补充：
抽象数据类型（Abstract Data Type [ADT]）：表示数学中抽象出来的一些操作的集合。
内存结构：内存中的结构，如：struct、特殊内存块...等等之类；

数组和链表的区别和优缺点：

数组是一种连续存储线性结构，元素类型相同，大小相等

数组的优点：
存取速度快

数组的缺点：
事先必须知道数组的长度
插入删除元素很慢
空间通常是有限制的
需要大块连续的内存块
插入删除元素的效率很低

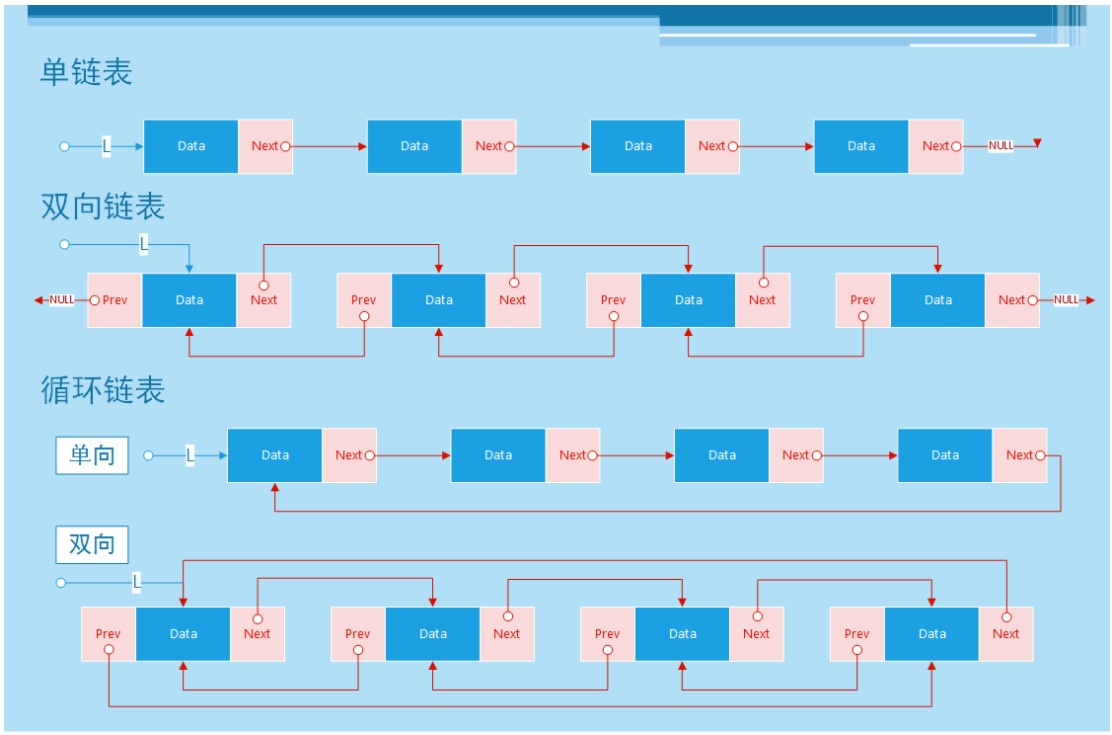
链表是离散存储线性结构
n 个节点离散分配，彼此通过指针相连，每个节点只有一个前驱节点，每个节点只有一个后续节点，首节点没有前驱节点，尾节点没有后续节点。

链表优点：
空间没有限制
插入删除元素很快

链表缺点：
存取速度很慢

2、链表共分几类?

链表常用的有 3 类： 单链表、双向链表、循环链表。



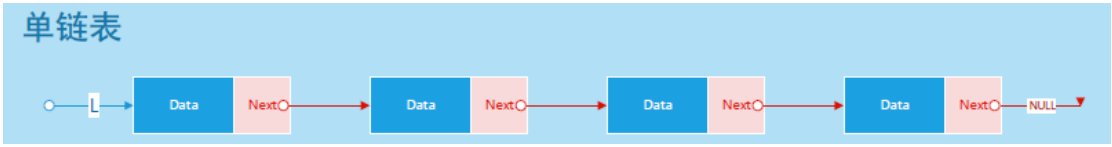
链表.png

链表的核心操作集有 3 种：插入、删除、查找(遍历)

单链表

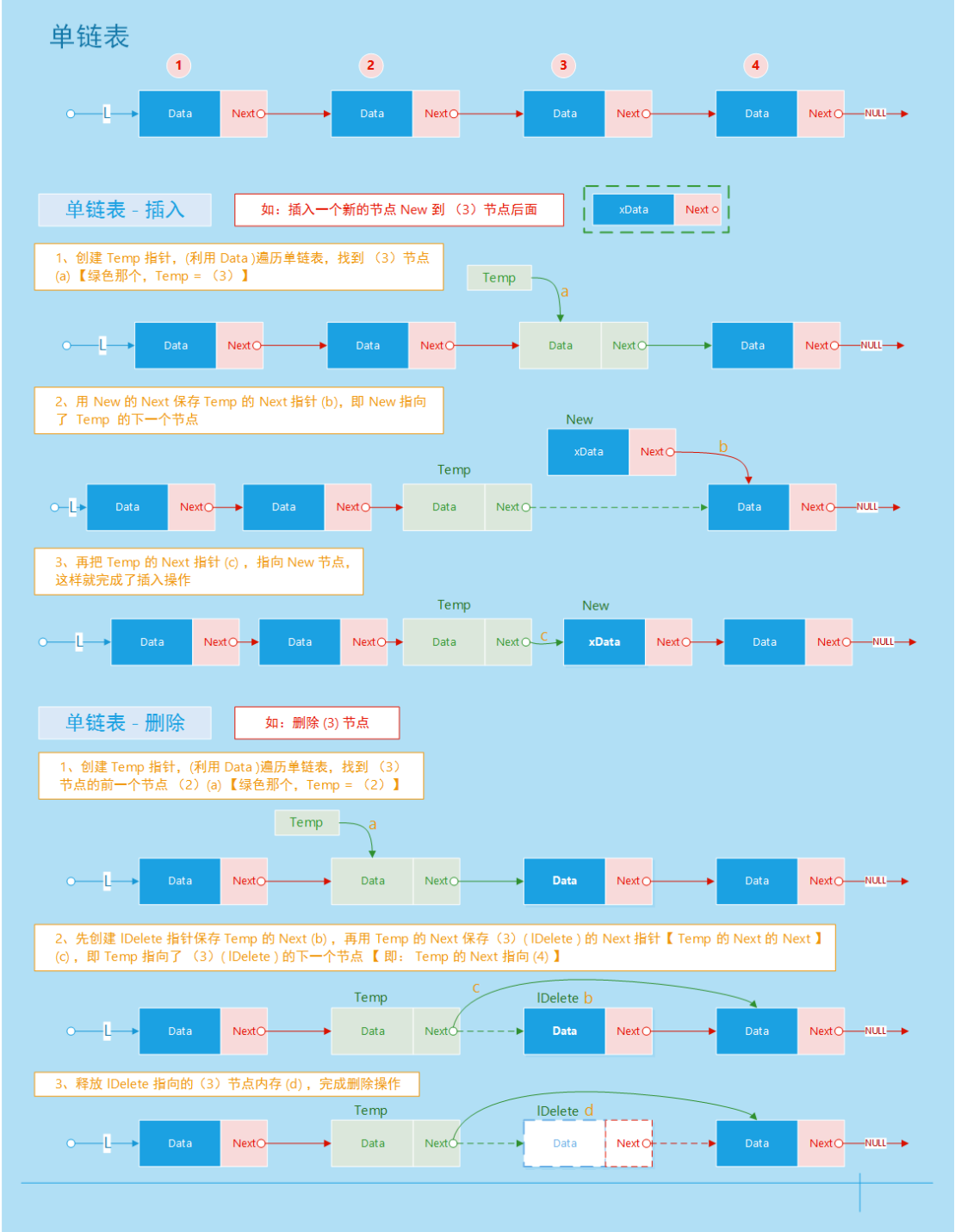
单链表 [Linked List]：由各个内存结构通过一个 Next 指针链接在一起组成，每一个内存结构都存在后继内存结构(链尾除外)，内存结构由数据域和 Next 指针域组成。

单链表实现图示：



解析：

Data 数据 + Next 指针，组成一个单链表的内存结构 ；
第一个内存结构称为 链头，最后一个内存结构称为 链尾；
链尾的 Next 指针设置为 NULL [指向空]；
单链表的遍历方向单一（只能从链头一直遍历到链尾）
单链表操作集：

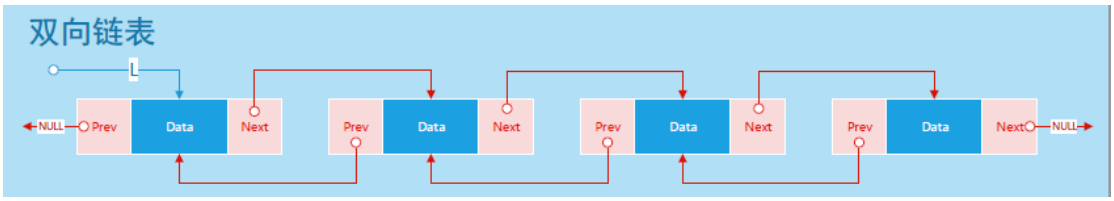


单向链表-操作.png

双向链表

双向链表 [Double Linked List]：由各个内存结构通过指针 Next 和指针 Prev 链接在一起组成，每一个内存结构都存在前驱内存结构和后继内存结构(链头没有前驱，链尾没有后继)，内存结构由数据域、Prev 指针域和 Next 指针域组成。

双向链表实现图示：



解析:

Data 数据 + Next 指针 + Prev 指针，组成一个双向链表的内存结构;

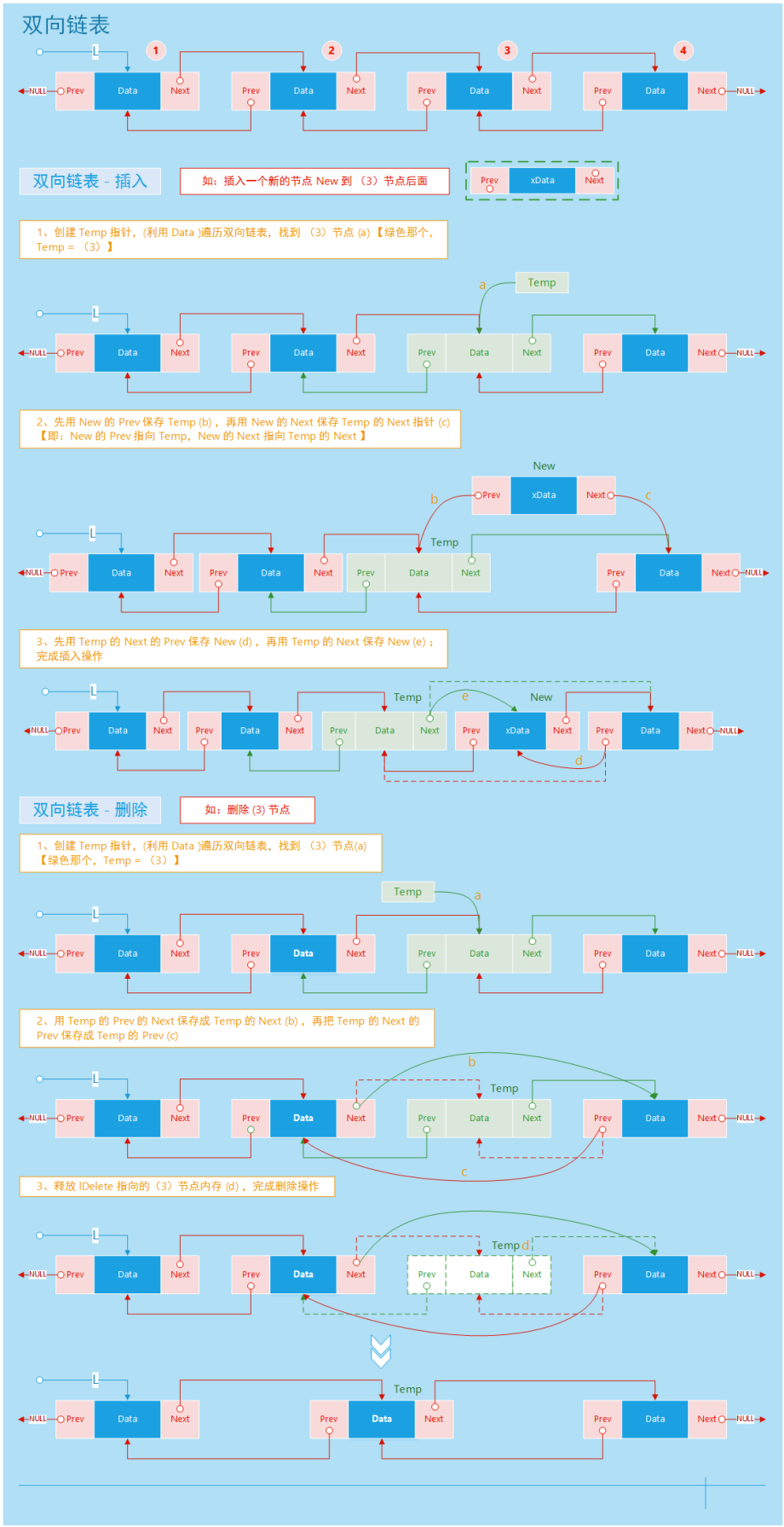
第一个内存结构称为 链头，最后一个内存结构称为 链尾;

链头的 Prev 指针设置为 NULL， 链尾的 Next 指针设置为 NULL;

Prev 指向的内存结构称为 前驱， Next 指向的内存结构称为 后继;

双向链表的遍历是双向的，即如果把从链头的 Next 一直到链尾的[NULL] 遍历方向定义为正向，那么从链尾的 Prev 一直到链头 [NULL]遍历方向就是反向;

双向链表操作集:


双向链表
L
1
2
3
4
NULL
Prev
Data
Next
双向链表 - 插入
如：插入一个新的节点 New 到 （3）节点后面
xData
1、创建 Temp 指针，(利用 Data)遍历双向链表，找到 （3）节点 (a)【绿色那个，Temp = （3）】
Temp
a
2、先用 New 的 Prev 保存 Temp (b)，再用 New 的 Next 保存 Temp 的 Next 指针 (c)【即：New 的 Prev 指向 Temp，New 的 Next 指向 Temp 的 Next 】
New
b
c
3、先用 Temp 的 Next 的 Prev 保存 New (d)，再用 Temp 的 Next 保存 New (e)；完成插入操作
e
d
双向链表 - 删除
如：删除 (3) 节点
1、创建 Temp 指针，(利用 Data)遍历双向链表，找到 （3）节点(a)【绿色那个，Temp = （3）】
2、用 Temp 的 Prev 的 Next 保存成 Temp 的 Next (b)，再把 Temp 的 Next 的 Prev 保存成 Temp 的 Prev (c)
3、释放 IDelete 指向的（3）节点内存 (d)，完成删除操作

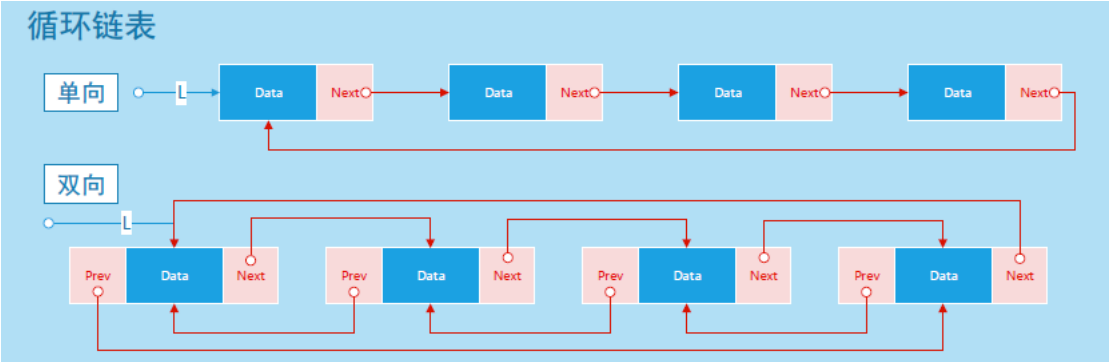
双向链表-操作.png

循环链表

单向循环链表 [Circular Linked List]：由各个内存结构通过一个指针 Next 链接在一起组成，每一个内存结构都存在后继内存结构，内存结构由数据域和 Next 指针域组成。

双向循环链表 [Double Circular Linked List]：由各个内存结构通过指针 Next 和指针 Prev 链接在一起组成，每一个内存结构都存在前驱内存结构和后继内存结构，内存结构由数据域、Prev 指针域和 Next 指针域组成。

循环链表的单向与双向实现图示：



解析：

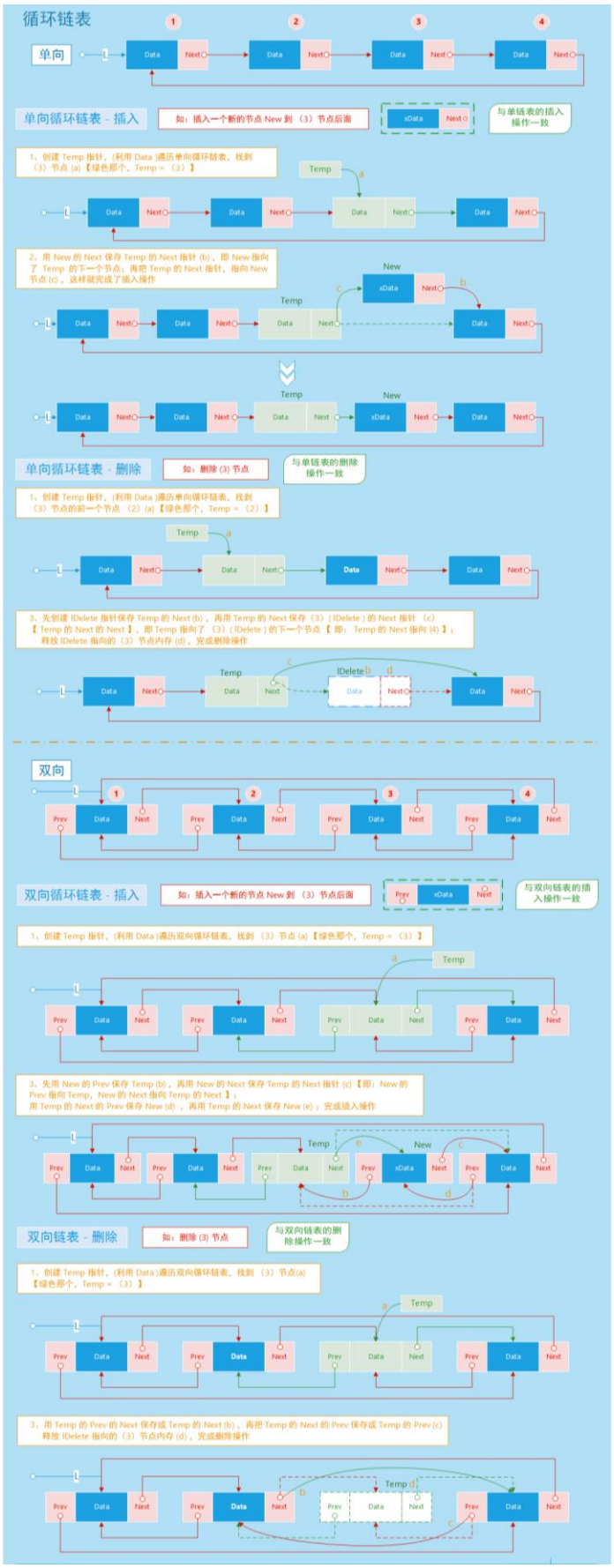
循环链表分为单向、双向两种；

单向的实现就是在单链表的基础上，把链尾的 Next 指针直接指向链头，形成一个闭环；

双向的实现就是在双向链表的基础上，把链尾的 Next 指针指向链头，再把链头的 Prev 指针指向链尾，形成一个闭环；

循环链表没有链头和链尾的说法，因为是闭环的，所以每一个内存结构都可以充当链头和链尾；

循环链表操作集：


循环链表
单向
Data
Next
单向循环链表 - 插入
如：插入一个新的节点 New 到 （3）节点后面
xData
与单链表的插入操作一致
Temp
New
单向循环链表 - 删除
如：删除 (3) 节点
与单链表的删除操作一致
IDelete
双向
Prev
双向循环链表 - 插入
与双向链表的插入操作一致
双向链表 - 删除
与双向链表的删除操作一致

循环链表-操作.png