**线性表:**全名为线性存储结构。使用线性表存储数据的方式可以这样理解，即“把所有数据用一根线儿[串](http://data.biancheng.net/view/175.html" \t "http://data.biancheng.net/view/_blank)起来，再存储到物理空间中”。

## 顺序存储结构和链式存储结构(线性表存储结构可细分为顺序存储结构和链式存储结构。)

1.将数据依次存储在连续的整块物理空间中，这种存储结构称为顺序存储结构（简称[顺序表](http://data.biancheng.net/view/158.html" \t "http://data.biancheng.net/view/_blank)）；

2.数据分散的存储在物理空间中，通过一根线保存着它们之间的逻辑关系，这种存储结构称为链式存储结构（简称[链表](http://data.biancheng.net/view/160.html" \t "http://data.biancheng.net/view/_blank)）；

## 线性表常用术语

数据结构中，一组数据中的每个个体被称为“数据元素”（简称“元素”）。例如，图 1 显示的这组数据，其中 1、2、3、4 和 5 都是这组数据钟的一个元素。  
  
另外，对于具有“一对一”逻辑关系的数据，我们一直在用“某一元素的左侧（前边）或右侧（后边）”这样不专业的词，其实线性表中有更准确的术语：

某一元素的左侧相邻元素称为“直接前驱”，位于此元素左侧的所有元素都统称为“前驱元素”；

某一元素的右侧相邻元素称为“直接后继”，位于此元素右侧的所有元素都统称为“后继元素”；

以图 1 数据中的元素 3 来说，它的直接前驱是 2 ，此元素的前驱元素有 2 个，分别是 1 和 2；同理，此元素的直接后继是 4 ，后继元素也有 2 个，分别是 4 和 5。如图 4 所示：



**顺序表**

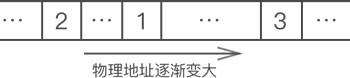
全名顺序存储结构，是[线性表](http://data.biancheng.net/view/157.html" \t "http://data.biancheng.net/view/_blank)的一种。通过《[线性表](http://data.biancheng.net/view/157.html" \t "http://data.biancheng.net/view/_blank)》一节的学习我们知道，线性表用于存储逻辑关系为“一对一”的数据，顺序表自然也不例外。

顺序表存储数据时，会提前申请一整块足够大小的物理空间，然后将数据依次存储起来，存储时做到数据元素之间不留一丝缝隙。  
  
例如，使用顺序表存储集合 {1,2,3,4,5}，数据最终的存储状态如[图](http://data.biancheng.net/view/200.html" \t "http://data.biancheng.net/view/_blank) 1 所示：



**链表**

别名链式存储结构或单链表，用于存储逻辑关系为 "一对一" 的数据。与[顺序表](http://data.biancheng.net/view/158.html" \t "http://data.biancheng.net/view/_blank)不同，链表不限制数据的物理存储状态，换句话说，使用链表存储的数据元素，其物理存储位置是随机的。  
  
例如，使用链表存储 {1,2,3}，数据的物理存储状态如[图](http://data.biancheng.net/view/200.html" \t "http://data.biancheng.net/view/_blank) 1 所示：



数据元素随机存储，并通过指针表示数据之间逻辑关系的存储结构就是链式存储结构。

**链表的节点**

链表中每个数据的存储都由以下两部分组成：

1. 数据元素本身，其所在的区域称为数据域；
2. 指向直接后继元素的指针，所在的区域称为指针域；

即链表中存储各数据元素的结构如图 3 所示：

IMG_256  
图 3 节点结构

图 3 所示的结构在链表中称为节点。也就是说，链表实际存储的是一个一个的节点，真正的数据元素包含在这些节点中，如图 4 所示：

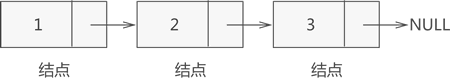


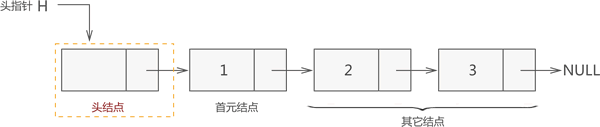
图 4 链表中的节点

## 头节点，头指针和首元节点

其实，图 4 所示的链表结构并不完整。一个完整的链表需要由以下几部分构成：

1. 头指针：一个普通的指针，它的特点是永远指向链表第一个节点的位置。很明显，头指针用于指明链表的位置，便于后期找到链表并使用表中的数据；
2. 节点：链表中的节点又细分为头节点、首元节点和其他节点：
   * 头节点：其实就是一个不存任何数据的空节点，通常作为链表的第一个节点。对于链表来说，头节点不是必须的，它的作用只是为了方便解决某些实际问题；
   * 首元节点：由于头节点（也就是空节点）的缘故，链表中称第一个存有数据的节点为首元节点。首元节点只是对链表中第一个存有数据节点的一个称谓，没有实际意义；
   * 其他节点：链表中其他的节点；

因此，一个存储 {1,2,3} 的完整链表结构如图 5 所示：

  
图 5 完整的链表示意图

注意：链表中有头节点时，头指针指向头节点；反之，若链表中没有头节点，则头指针指向首元节点。