# **单链表**

## **一、什么是链表**

**链表是数据结构中线性表的一种，其中的每个元素实际上是一个单独的结构体对象，而所有对象都通过每个元素中的指针链接在一起。它是以结构体为节点，将一个结构体看成数据域和指针域两个部分，数据域用于存储数据，指针域用于连接下一个节点，链表中每个结构体对象叫做节点，其中第一个数据节点叫做链表的首元节点；如果第一个节点不用于存储数据，只用于代表链表的起始点，则这个节点称为链表的头节点。**

## **二、链表的特点**

**1、链表没有固定的长度，可以自由增加节点**

**2、链表能够实现快速的插入删除数据，也就是可以快速的插入和删除链表中的节点**

1. **与数组类似，链表也是一种线性数据结构**
2. **链表的尾结点的后继必定指向空**

**链表和数组的区别：**

**数组和顺序表是顺序存储的，也就是内存是连续的；而链表是通过指针将不连续的内存连接起来，实现链式存储的。**

## **三、单链表结构体的声明**

**typedef int Type; //数据类型 通过取别名的形式进行灵活使用**

**struct Node{ //单链表节点结构体的声明**

**Type data; //链表节点的数据域，用于存储数据**

**struct Node \*next; //链表节点的指针域，用于指向和连接下一个节点**

**};**

**struct LinkList{ //单链表结构体的声明**

**struct Node \*head; //链表头节点的指针域，用于指向链表的开头**

**struct Node \*next; //链表尾节点的指针域，用于指向链表的末尾**

**int lenth; //链表的长度**

**};**

## **单链表的结构示意图**

## 

## **五、单链表的设计与功能实现**

**1、创建单链表**

1. **链接单链表节点**

**3、单链表的打印输出**

**4、单链表节点的插入**

**5、单链表节点的删除**

**6、获取单链表指定位置上的数据**