# 面试官：说说你了解的js数据结构？

## 什么是数据结构？

数据结构是计算机存储、组织数据的方式。 数据结构意味着接口或封装：一个数据结构可被视为两个函数之间的接口，或者是由数据类型联合组成的存储内容的访问方法封装。

我们每天的编码中都会用到数据结构 数组是最简单的内存数据结构 下面是常见的数据结构： 1. 数组（Array） 2. 栈（Stack） 3. 队列（Queue） 4. 链表（Linked List） 5. 字典 6. 散列表（Hash table） 7. 树（Tree） 8. 图（Graph） 9. 堆（Heap）

## 数组（Array）

数组是最最基本的数据结构，很多语言都内置支持数组。 数组是使用一块连续的内存空间保存数据，保存的数据的个数在分配内存的时候就是确定的。

在日常生活中，人们经常使用列表：待办事项列表、购物清单等。

而计算机程序也在使用列表，在下面的条件下，选择列表作为数据结构就显得尤为有用： 数据结构较为简单 不需要在一个长序列中查找元素，或者对其进行排序 反之，如果数据结构非常复杂，列表的作用就没有那么大了。

## 栈（Stack）

栈是一种遵循后进先出（LIFO）原则的有序集合 在栈里，新元素都接近栈顶，旧元素都接近栈底。 每次加入新的元素和拿走元素都在顶部操作

## 队列（Queue）

队列是遵循先进先出（FIFO，也称为先来先服务）原则的一组有序的项 队列在尾部添加新元素，并从顶部移除元素 最新添加的元素必须排在队列的末尾

## 链表（Linked List）

链表也是一种列表，已经设计了数组，为什么还需要链表呢？ JavaScript中数组的主要问题时，它们被实现成了对象， 与其他语言（比如C++和Java）的数组相对，效率很低。 如果你发现数组在实际使用时很慢，就可以考虑使用链表来代替它。

使用条件： 链表几乎可以用在任何可以使用一维数组的情况中。 如果需要随机访问，数组仍然是更好的选择。

## 字典

字典是一种以键-值对存储数据的数据结构，js中的Object类就是以字典的形式设计的。JavaScript可以通过实现字典类，让这种字典类型的对象使用起来更加简单，字典可以实现对象拥有的常见功能，并相应拓展自己想要的功能，而对象在JavaScript编写中随处可见，所以字典的作用也异常明显了。

## 散列表

也称为哈希表，特点是在散列表上插入、删除和取用数据都非常快。 为什么要设计这种数据结构呢？ 用数组或链表存储数据，如果想要找到其中一个数据，需要从头进行遍历，因为不知道这个数据存储到了数组的哪个位置。

散列表在JavaScript中可以基础数组去进行设计。 数组的长度是预先设定的，所有元素根据和该元素对应的键，保存在数组的特定位置，这里的键和对象的键是类型的概念。 使用散列表存储数组时，通过一个散列函数将键映射为一个数字，这个数字的范围是0到散列表的长度。

即使使用一个高效的散列函数，依然存在将两个键映射为同一个值得可能，这种现象叫做碰撞。常见碰撞的处理方法有：开链法和线性探测法（具体概念有兴趣的可以网上自信了解） 使用条件： 可以用于数据的插入、删除和取用，不适用于查找数据 