如果让一个数组不能配置，就不能删除它。如果不能删除它，length的属性不能设置为小于不可配置元素的索引值。

数组索引仅仅是对象属性名的一种特殊类型，这意味着javascript数组没有越界的错误概念，查询一个不存在的值并不会报错，只会得到undefined值。类似于对象，对象同样存在这种情况

数组既然是对象，那么他可以从原型中继承元素，在ECMAScript5中数组可以定义getter和setter方法。如果数组确实继承了元素或使用元素的getter和setter方法，应该其他它使用费优化的代码路径：访问这种数组的元素的时间会与常规对象属性的查找时间相近。

稀疏数组就是包含从0开始的不连续索引的数组，通常，数组的length属性值代表数组中元素的个数。稀疏数组length值大于元素个数。可以用Array()构造函数或简单的指定数组索引值大于数组长度来创建稀疏数组。在数组直接量中省略值时不会创建稀疏数组。省略的元素是存在的其值为undefined；

数组有length属性表示数组个数，其值比数组中最大的索引值大1；

数组的特殊行为1:如果为一个数组元素赋值，它的索引i大于或等于现有数组长度时，length属性的值将设为i+1

2：设置length属性为一个小于当前长度的非负数整数n时，当前数组中那些索引值大于或等于n的元素将从中删除。

将数组的length值设置为大于其当前的长度。实际上这不会向数组中添加元素，只是在尾部创建一个空的区域。

如果让数组元素不可配置，就不能删除它，如果不能删除。Length属性不能设置为小于不可配置元素的索引值.

数组特性

* 当有新的元素添加到列表中时，自动更新length属性
* 设置length为一个较小值时截断数组
* 从Array.prototype中继承一些有用的方法
* 其类属性为“Array”

Javascript数组方法特意定义为通用的，因此他们不仅应用在真正的数组而且在类数组对象上都能正确的工作。在ECMAScript5中，所有的数组方法都是通用的，在ECMAScript3中除了toString()和toLocalString()以外其他的方法也是通用的(concat()方法是一个特例，虽然可以用在类数组对象上，但是它没有将那个对象扩充进返回的数组中)既然数组对象没有继承自Array.prototype那就不能在他们上面直接调用数组方法，可以间接的使用Function.call方法例如Array.prototype.join.call()；