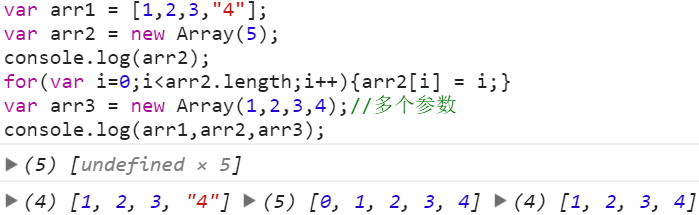
**第十八章 Array数组**

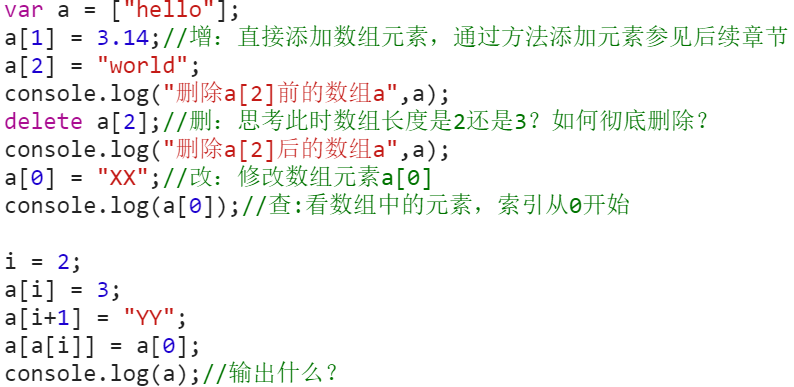
**创建数组的方式**

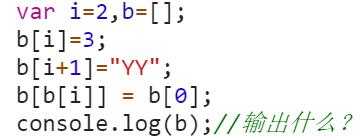
通过字面量的方式直接创建，直接量中的值可以是任意的表达式

通过Array构造函数来创建数组对象，注意传递的参数。



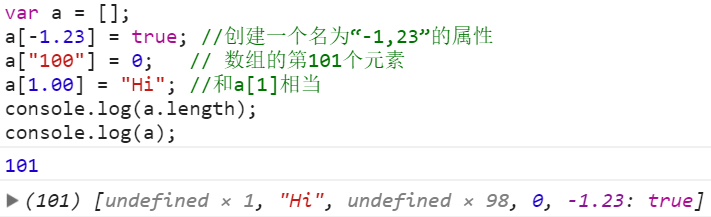
**数组元素的增删改查的基本操作**

****

****

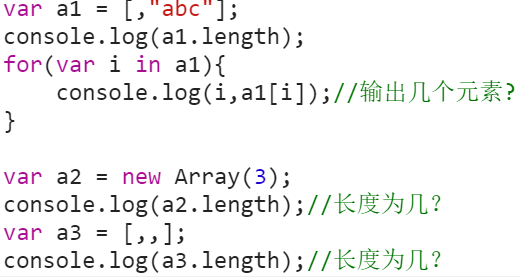
**数组相当于普通对象的特别之处**

- 数组是对象的特殊形式，可以为数组添加对象属性，对于0至2的32次方之外的数，将作为普  
 通对象的键来对待  
- 数组特别之处在于，当使用使用2的32次方以内的非负整数作为属性名时（包括类型转换的数  
 字），数组会自动维护其length属性，作为数组的元素，而不是数组对象的属性



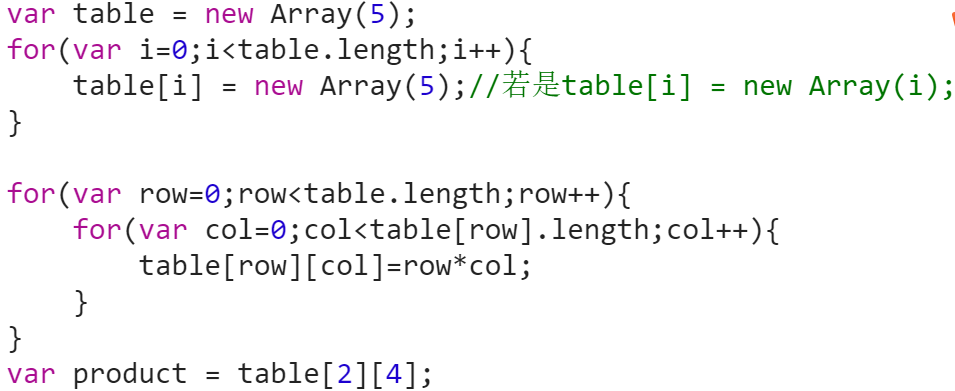
**稀疏数组**

* - 稀疏数组是包含从0开始的不连续索引的数组，length值大于实际定义的元素的个数  
  - 遍历稀疏数组时，注意的跳过无元素项的问题



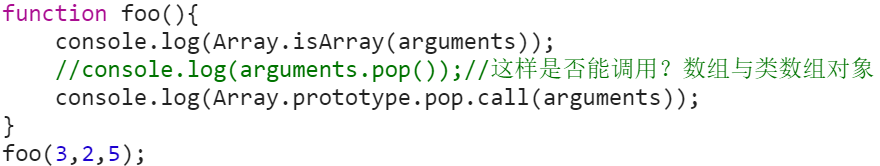
**多维数组（矩形数组、交错数组）**

* - JS中可以通过包含数组的数组来模拟多维数组



**数组静态方法（构造器函数对象的方法）**- Array.from(...) 、Array.isArray(...) 、Array.of(...)等（具体参见实例）

**数组原型方法（添加和删除元素-破坏性）**- Array.prototype.shift() Array.prototype.unshift(elem1?,elem2?,...)  
- Array.prototype.pop() Array.prototype.push(elem1?,elem2?,...)  
- Array.prototype.splice(start,deleteCount?,elem1?,elem2?)



**数组原型方法（排序和颠倒元素顺序-破坏性）**- Array.prototype.reverse()  
- Array.prototype.sort(compareFunction？) //回调函数的写法，思考冒泡排序

**数组原型方法（合并、切分和连接-非破坏性）**  
- Array.prototype.concat(arr1?,arr2?,...)  
- Array.prototype.slice(begin?,end?) //注意参数的正负，注意不要和splice混淆了  
- Array.prototype.join(separator?) //注意返回的类型

**数组原型方法（值的查找-非破坏性）**- Array.prototype.indexOf(searchValue,startIndex?)  
- Array.prototype.lastIndexOf(searchElement,startIndex?) //注意方向和起始点

**数组原型方法（迭代-非破坏性-检测方法）**- Array.prototype.forEach(callback,thisValue?)   
- Array.prototype.every(callback,thisValue?) //若有不满足的，立即返回false，不再后续迭代  
- Array.prototype.some(callback,thisValue?) //若有满足的，立即返回true，不再后续迭代

**数组原型方法（迭代-非破坏性-转换方法）**- Array.prototype.map(callback,thisValue?)  
- Array.prototype.filter(callback,thisValue?)

**数组原型方法（迭代-非破坏性-归约方法）**  
- Array.prototype.reduce(element,initialValue?)  
- Array.prototype.reduceRight(callback,initialValue?)