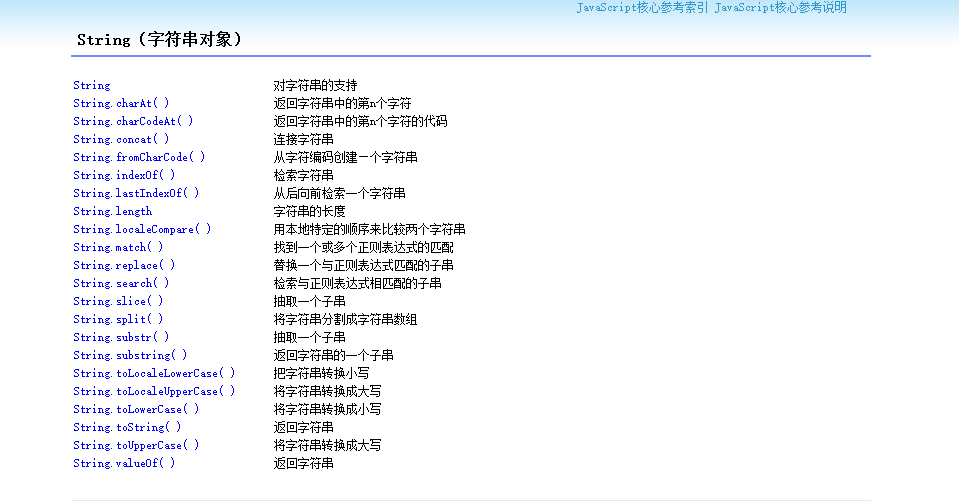
# ES5/ES6 String/Array方法

## string方法 ES5



var str = '1\*234567890123A';

console.log(str.charAt(8));//index

console.log(str.charCodeAt(9));//index

console.log(str.concat('iii'));//拼接

console.log(String.fromCharCode(49));//unicode编码

console.log(str.indexOf('2',3))//默认从0位置开始查找‘2’，第二个参数可以指定从哪开始；

console.log(str.lastIndexOf('23'));//求字串位置,默认从末尾开始

console.log(str.replace('1','\*\*'));//替换字串

console.log(str.localeCompare('12345678'));//比较大小

console.log(str.match(/1/g));//找到匹配项返回数组

console.log(str.slice(0,-4));//抽出字串

//一个新的字符串。包括字符串 stringObject 从 start 开始（包括 start）到 end 结束（不包括 end）为止的所有字符。

//结束可以使用负数

console.log(str.split('2'));

//以指定字符为间隔将字符串转化为数组

console.log(str.substr(0,4));//后面的参数是长度

console.log(str.substring(0,4));//后面的参数不允许是负数

console.log(str.toLocaleLowerCase());

//与 toLowerCase() 不同的是，toLocaleLowerCase() 方法按照本地方式把字符串转换为小写。只有几种语言（如土耳其语）具有地方特有的大小写映射，所有该方法的返回值通常与 toLowerCase() 一样

console.log(str.toLocaleUpperCase());

console.log(str.toLowerCase());

console.log(str.toUpperCase());

console.log(str.valueOf());

console.log(str.toString());

字符串也可以使用下标进行操作；

## string方法 ES6

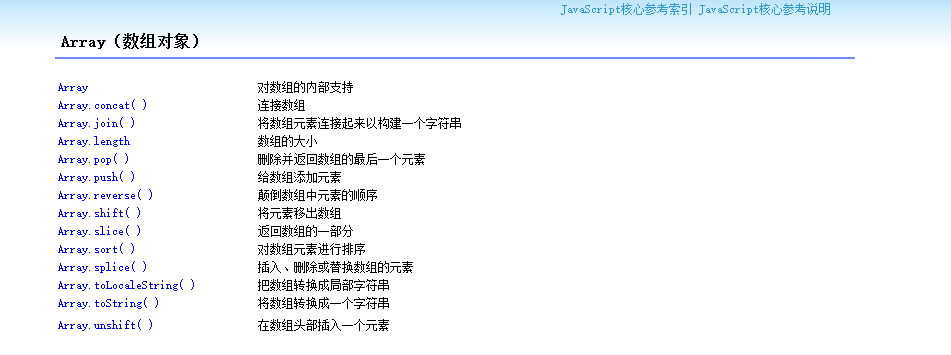
1. includes(str) : 判断是否包含指定的字符串

2. startsWith(str) : 判断是否以指定字符串开头

3. endsWith(str) : 判断是否以指定字符串结尾

4. repeat(count) : 重复指定次数

## Array方法 ES5

arr.sort(function(a,b){

return b - a;

})

语法：array.sort(fun)；参数fun可选。规定排序顺序。必须是函数。

注：如果调用该方法时没有使用参数，将按字母顺序对数组中的元素进行排序，说得更精确点，是按照字符编码的顺序进行排序。

如果想按照其他规则进行排序，就需要提供比较函数，该函数要比较两个值，然后返回一个用于说明这两个值的相对顺序的数字。比较函数应该具有两个参数 a 和 b，其返回值如下：

若 a 小于 b，在排序后的数组中 a 应该出现在 b 之前，则返回一个小于 0 的值。若 a 等于b，则返回 0。若 a 大于 b，则返回一个大于 0 的值。

简单点就是：比较函数两个参数a和b，返回a-b 升序，返回b-a 降序，原数组发生改变

## Array方法 ES5 + ES6

### ES5

1. Array.prototype.indexOf(value) : 得到值在数组中的第一个下标

2. Array.prototype.lastIndexOf(value) : 得到值在数组中的最后一个下标

3. Array.prototype.forEach(function(item, index){}) : 遍历数组

arr = [1,23];

arr.forEach(function(item,index){

console.log(item);

})

4. Array.prototype.map(function(item, index){}) : 遍历数组返回一个新的数组，返回加工之后的值

map() 方法返回一个新数组，数组中的元素为原始数组元素调用函数处理后的值。

map() 方法按照原始数组元素顺序依次处理元素。

注意： map() 不会对空数组进行检测。

注意： map() 不会改变原始数组。

console.log(arr.map(function(item,index){

return item \* 2;

}))

5. Array.prototype.filter(function(item, index){}) : 遍历过滤出一个新的子数组， 返回条件为true的值

console.log(arr.filter(function(item,index){

return item >= 2;

}))

### ES6

1. Array.from(v) : 将伪数组对象或可遍历对象转换为真数组

var arr1 = '12345';

console.log(Array.from(arr1));

1. Array.of(v1, v2, v3) : 将一系列值转换成数组

console.log(Array.of(1,2,3)); //console.log(new Array(1,2,3));

1. find(function(value, index, arr){return true}) : 找出第一个满足条件返回true的元素

console.log(arr.find(function(item,index){

return item > 10;

}));

4. findIndex(function(value, index, arr){return true}) : 找出第一个满足条件返回true的元素下标

console.log(arr.findIndex(function(item,index){

return item > 10;

}));

所有的方法都要注意三要素；

## 强化练习（面试题）

1. 字符串反转 （使用字符串和数组的方法实现）

2. 打印字符串中出现字母最多的次数 \*\*\*\*\*

作业

1. 熟悉字符串的方法（三要素）

2. 熟悉数组的方法（三要素）

3. 字符串课上面试题

4. 数组课上面试题