

JavaScript 基础第六天

学习目标

- ☐ 数组方法，重要，做项目的时候经常需要操作数组。
- ☐ 掌握数组的 `push()`、`pop()` 的使用
- ☐ 掌握数组的 `concat()`、`slice()`、`splice()` 的使用
- ☐ 掌握数组的 `indexOf()`、`join()`、`reverse()` 的使用
- ☐ 数值方法 `toFixed()`
- ☐ 字符串方法
- ☐ 掌握字符串的 `concat()`、`substring()` 的使用
- ☐ 掌握字符串的 `replace()`、`split()` 的使用

如何查阅手册

目标

学习通过查阅查阅对象的方法

W3School 截图参考

进入 Array 对象完整手册

JS 指南

JavaScript 对象

JS 对象简介

JS 字符串

JS 日期

JS 数组

JS 逻辑

JS 算数

JS RegExp

JS HTML DOM

JavaScript 高级

JS 浏览器

JS Cookies

JS 验证

JS 动画

JS 图像地图

文字数组 - `sort()`

如何使用 `sort()` 方法从字面上对数组进行排序。

数字数组 - `sort()`

如何使用 `sort()` 方法从数值上对数组进行排序。

完整的 Array 对象参考手册

我们提供 [JavaScript Array 对象参考手册](#)，其中包括所有可用于数组对象的属性和方法。

该手册包含了对每个属性和方法的详细描述以及相关实例。

白名单

点击进入详情页面



Array 对象方法

FF: Firefox, IE: Internet Explorer

点击其中一个方法，进入到该方法的详情页面

方法	描述	FF	IE
concat()	连接两个或更多的数组，并返回结果。	1	4
join()	把数组的所有元素放入一个字符串。元素通过指定的分隔符进行分隔。	1	4
pop()	删除并返回数组的最后一个元素	1	5.5
push()	向数组的末尾添加一个或更多元素，并返回新的长度。	1	5.5
reverse()	颠倒数组中元素的顺序。	1	4
shift()	删除并返回数组的第一个元素	1	5.5
slice()	从某个已有的数组返回选定的元素	1	4
sort()	对数组的元素进行排序	1	4
splice()	删除元素，并向数组添加新元素。	1	5.5
toSource()	返回该对象的源代码。	1	-
toString()	把数组转换为字符串，并返回结果。	1	4
toLocaleString()	把数组转换为本地数组，并返回结果。	1	4
unshift()	向数组的开头添加一个或更多元素，并返回新的长度。	1	6
valueOf()	返回数组对象的原始值	1	4

详情页面有跟详细的说明

JavaScript push() 方法

JavaScript Array 对象参考手册

定义和用法

定义和用法就是这个方法的功能描述

push() 方法可向数组的末尾添加一个或多个元素，并返回新的长度。

语法

语法：就是这个方法正确使用知识，不按语法书写就可能报错

```
arrayObject.push(newelement1,newelement2,...,newelementX)
```

参数	描述
newelement1	必需。要添加到数组的第一个元素。
newelement2	可选。要添加到数组的第二个元素。
newelementX	可选。可添加多个元素。

返回值

看到有返回值，说明方法内部有 return，返回值可以通过赋值给某个变量

把指定的值添加到数组后的新长度。

说明

说明一般是细节补充

push() 方法可把它的参数顺序添加到 arrayObject 的尾部。它直接修改 arrayObject，而不是创建一个新的数组。push() 方法和 pop() 方法使用数组提供的先进后出栈的功能。

提示和注释

注释：该方法会改变数组的长度。

提示：要想数组的开头添加一个或多个元素，请使用 unshift() 方法。这里提示可用 unshift() 前添加数据

MDN 在线手册

MDN 搜索页面：<https://developer.mozilla.org/zh-CN/search>

搜索结果: "Array" | MDN

https://developer.mozilla.org/zh-CN/search?q=Array&topic=api&topic=css&topic=canvas&topic=html&topic=http&topic=js&topic=svg...

MDN web docs

技术 参考与指南 反馈

登录

搜索

Array 输入需要搜索的知识点

结果

找到 3084 篇匹配"Array"的文档，于 中文 (简体) 和 English (US)。已显示结果 1 至 10。 [只在 English \(US\) 中再次搜索](#)

文档

Array
Arrays are used to store multiple values in a single variable....An **array** is an ordered collection of data (either

Array [点击超链接跳转到数组对象的详情页](#)
JavaScript的 **Array** 对象是用于构造数组的全局对象，数组是类似于列表的高阶对象。
developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/Global_Objects/Array 已从 [Array \(中文 \(简体\)\)](#) 翻译

☐ 所有主题

主题

- ✓ API 和 DOM 1833
- ✓ CSS 80
- ✓ Canvas 81
- ☐ Firefox 78
- ☐ Firefox OS 152
- ✓ HTML 101
- ✓ HTTP 19
- ✓ JavaScript 1081
- ☐ MathML 6
- ✓ SVG 12
- ✓ Web Standards 106

小结

对象的方法都有很多，初次接触很容易记不住，通过手册查阅其实是最好的办法。

W3School: 适合入门查看

MDN: 适合深入学习，功能全，需要在线浏览。

数组对象的方法 *

目标

学习数组方法 `push` , `unshift` , `pop` , `shift` , `splice` , `reverse` , `join` , `forEach` 。

`includes()`

用于判断一个数组是否包含一个指定的数。包含true，不包含false

```
var prices = [500, 8123, 12];  
prices.include (2) ; //false
```

`join()`

可以把数组转换成字符串，并且每个数据之间可以用指定的字符串分隔。返回新的字符串

特殊用法：

如果想让数组的数据之间紧密相连，就传入一个 空字符串。

push, unshift, pop, shift

后添加	<code>.push(数据1, 数据2...)</code> 返回数组的新长度	可添加一个或多个	<code>.splice(数组.length, 添加的数据)</code>
后删除	<code>.pop()</code> 返回被删除的数据	只能删除一个	<code>.splice(-1,1)</code>
前添加	<code>.unshift(数据1, 数据2...)</code> 返回数组的新长度	可添加一个或多个	<code>.splice(0,0,添加的数据)</code>
前删除	<code>.shift()</code> 返回被删除的数据	只能删除一个	<code>.splice(0,1)</code>

.splice()

参数1(必填)：要删除元素的索引值，第一位是 0，最后一位是 -1

参数2(必填)：删除的个数，个数为 0 代表不删除

参数3：添加的元素，参数3往后都代表添加的元素

返回值：以数组格式，返回的是被删除的数据

清空数组：参数写上一个0即可

注意事项：

- `.pop()` 和 `.shift()` 删除返回的一个数据，而 `.splice()` 删除返回的 数组 格式。
- 一个参数都不传也不会报错，其实三个参数都是可选参数。填写一个参数，就是从指定参数开始截取

.slice()

参数1：选取的开始索引值

参数2：选取的结束索引值(不包括该元素)

注意事项：

- 如果只传入一个参数，代表从开始索引值直接选取到最后。
- 该功能是选取数据，不影响原数组。
- 一个参数都不写，返回值是一个全新的数组，可以用来复制一份新数组，又不影响原数组。

`.splice()` 和 `.slice()` 区别

`splice` 主要用于删除, `slice` 用于提取

	<code>splice *</code>	<code>slice</code>
主要功能	删除, 可添加数据	提取
是否影响原数组	是, 原数组受影响	否, 不影响原数组
返回值	返回被删除的数据	返回提取那部分数据
参数区别	数组.splice(开始索引值, 个数)	数组.slice(开始索引值, 结束索引值)
是否能添加数组	可以添加, 第三个参数往后都是添加	仅仅提取, 不能修改原数组, 无法添加

`concat()`

用于拼接两个数组, 返回一个全新的数组

语法:

`[]`.concat(数组1, 数组2) 常用

数组1.concat(数组2)

不会影响原数组

`reverse()`

反转数组, 会改变原数组

语法:

数组.reverse()

`.sort()`

对指定数组进行排序

语法:

```
数组名称.sort([排序函数])  
// 默认排序规则是比较第一位谁小, 就排到前面, 第一位比较不出来就比较第二位。
```

截图参考

```
<script type="text/javascript">

function sortNumber(a,b)
{
    return a - b
}

var arr = new Array(6)
arr[0] = "10"
arr[1] = "5"
arr[2] = "40"
arr[3] = "25"
arr[4] = "1000"
arr[5] = "1"

document.write(arr + "<br />")
document.write(arr.sort(sortNumber))

</script>
```

排序函数，需要传入两个参数

内部需要返回 两个数相减的差值

从小到大数值排序需要传入排序函数：

```
// 从小到大排序
function sortNumber(a, b) {
    return a - b;
}

// 从大到小排序
function sortNumber(a, b) {
    return b - a;
}

// 调用排序函数进行排序
数组名称.sort(排序函数)
```

注意：sort 函数会影响原数组的数据，直接对原数组进行排序。

.forEach() 遍历 **

这个方法专门用于遍历数组。

用法：

```
数组.forEach(  
    function(item,index){  
        item  就是数组里面的每个元素  
        index 就是每个索引  
    }  
);
```

但凡是能够使用for循环遍历数组全部数据的时候，都可以使用 forEach 来代替。

`.filter()` 过滤

这个方法用于筛选出数组中满足条件的元素，并组成一个新的数组返回的

```
const array = [1500,1200,2000,2100,1800];  
const arr = array.filter(function(item){  
    return item < 2000; // 这个代码的意思，就是筛选出array数组里面满足 < 2000 的元素，并返回添加到一个新  
    数组里面  
});  
console.log(arr); // [1500, 1200, 1800]
```

`.indexOf()` 返回索引值

数组名称.indexOf(数据);
查找结果只有两种情况：
1. 查找成功，返回当前查找数据的索引值
2. 找不到返回 -1

数组名称.indexOf(数据,从哪个索引值开始);
默认是从0开始找，找到最近的指定数据，并放回索引值
传入索引值，就是从传入的索引值开始往后找，找到最近的指定数据，并放回索引值

数组小结

会影响原数组的	功能翻译	备注
<code>pop()</code>		
<code>push()</code>		
<code>shift()</code>		
<code>unshift()</code>		
<code>splice()</code>		
<code>reverse()</code>		
<code>sort()</code>		

不会影响原数组	功能翻译	备注
<code>slice()</code>	主要用来提取数组	<code>.slice(开始索引值, 结束索引值)</code>
<code>concat()</code>	返回连接数组	
<code>join()</code>	返回数组分隔成字符串	

String 对象的方法

目标


学习字符串属性方法：`length`，`indexOf`，`slice`，`replace`


字符串对象伪数组特征

字符串其实就是多个字符串联起来，也是有索引有长度的，字符串对象是伪数组格式。

▼ `String {"你好呀!"}`  ← "object" 字符串对象

0: "你"
1: "好"
2: "呀"
3: "!"
length: 4

 索引值

 长度

► `__proto__`: String
`[[PrimitiveValue]]`: "你好呀!"

得到字符串的长度: 字符串.length

根据索引得到字符: 字符串[索引] 或者 字符串.charAt(索引)

`.indexOf()`

字符串索引值查找

字符串.indexOf()

这个方法是根据某个字符, 查找这个字符在字符串里面的索引, 如果存在, 返回索引, 如果不存在, 返回-1

```
var str = "abc";  
console.log(str.indexOf('c')); // 2  
console.log(str.indexOf('z')); // -1
```

`.replace()`

字符串替换 - 正则表达式的时候讲解

字符串.replace(旧值, 新值)

这个方法是将字符串中的某些字符替换成新的字符

参数:

第一个参数: 需要替换的字符串, 一般写正则表达式, 数据需要用两个斜杠包起来: 如: `/你好/`

第二个参数: 新字符串用于替换原字符串

补充:

字符串.replace(`/字符串/g`, "新字符串");

`g` 代表 `global`, 全局替换。

`.slice()`

字符串提取其实有三个方法: `.slice()` `.substring()` `.substr()`, 主要掌握其中 `.slice()` 即可。

字符串.slice(开始索引值, 结束索引值)
字符串.substring(开始索引值, 结束索引值)
字符串.substr(开始索引值, 提取个数)

用来提取字符串里面的某一部分字符

```
// 字符串.substring(从哪里开始截取,到哪里结束)
// 注意点,得到的结果,是不包含结束索引的
var str = 'abcdef';
// 截取de
console.log(str.substring(3,5)); // 结果为: de
```

注意:

- substring 功能和 slice 功能几乎一样, 截取的时候不包含结束索引值。
- slice 索引允许负数。

字符串不可变 - 特征

JavaScript 的字符串是不可变的 (immutable) , String 类定义的方法都不能改变字符串的内容。

所有的字符串方法都是 返回的是全新的字符串, 而不是修改原始字符串。

查手册

对象的属性和方法都很多, 学会如何查手册, 学习对象的属性和方法。

Math 内置对象方法

方法名	功能解释	备注
<code>.random()</code>	获取随机数	获取 0 ~ 1 的随机小数
	四舍五入取整	
	向下取整	
	向上取整	
	最大值	
	最小值	

Date 对象方法

方法名	功能解释	备注
<code>.getFullYear()</code>	获取年	
	获取月	
	获取日	
	获取时	
	获取分	
	获取秒	
	获取星期	
	获取时间戳	

Array 对象方法

方法名	功能解释	备注
影响原数组的方法	-	-
<code>.pop()</code>		
<code>.push()</code>		
<code>.shift()</code>		
<code>.unshift()</code>		
<code>.splice()</code>		
<code>.sort()</code>		
<code>.reverse()</code>		
不影响原数组的方法,	-	返回值接收结果
<code>.slice()</code>		
<code>.concat()</code>		
<code>.join()</code>		
<code>.indexOf()</code>		
其他数组方法		
<code>.forEach()</code>		

Number 对象方法

方法名	功能解释	备注
<code>.toFixed()</code>		

String 对象方法

方法名	功能解释	备注
<code>.split()</code>		
<code>.indexOf()</code>		
<code>.slice()</code>		
<code>.substring()</code>		
<code>.replace()</code>		
<code>.toLowerCase()</code>		
<code>.toUpperCase()</code>		

注意：JavaScript 的字符串是不可变的，String 类定义的方法都不能改变字符串的内容。