

JS Array（数组）对象

数组是值的有序集合，数组中的每个值称为一个元素，每个元素在数组中都有一个数字位置，称为索引，索引从 0 开始，依次递增。在 JavaScript 中，您可以使用 Array 对象定义数组，此外，Array 对象中还提供了各种有关数组的属性和方法。

创建 Array 对象的语法格式如下：

```
var arr = new Array(values);
var arr = Array(values);
```

其中，values 为数组中各个元素组成的列表，多个元素之间使用逗号分隔。

示例代码如下：

```
01. var fruits = new Array( "apple", "orange", "mango" );
02. console.log(fruits); // 输出: ["apple", "orange", "mango"]
```

提示：在使用 new Array() 来定义数组时，如果只提供一个数值参数，那么这个数值将用来表示数组的初始长度，例如 new Array(5) 表示定义一个长度为 5 的数组。JavaScript 中，数组允许的最大长度为 $2^{32}-1$ ，即 4294967295。

除了可以使用 Array() 函数来定义数组外，您也可以直接使用方括号 [] 来定义数组，[] 中为数组中的各个元素，多个元素之间使用逗号 , 进行分隔。示例代码如下：

```
01. var fruits = [ "apple", "orange", "mango" ];
02. console.log(fruits); // 输出: (3) ["apple", "orange", "mango"]
```

您可以通过数组的索引来访问数组中的各个元素，示例代码如下：

```
01. var fruits = [ "apple", "orange", "mango" ];
02.
03. document.write(fruits[0] + "<br>"); // 输出: apple
04. document.write(fruits[1] + "<br>"); // 输出: orange
05. document.write(fruits[2] + "<br>"); // 输出: mango
```

Array 对象中的属性

下表中列举了 Array 对象中提供的属性及其描述信息：

属性	描述
constructor	返回创建数组对象的原型函数
length	设置或返回数组中元素的个数

prototype	通过该属性您可以向对象中添加属性和方法
-----------	---------------------

示例代码如下：

```
01. var cars = new Array("Saab", "Volvo", "BMW");
02.
03. Array.prototype.name = null;
04. cars.name = "JavaScript";
05. document.write(cars.constructor + "<br>");           // 输出: function Array() {
    [native code] }
06. document.write(cars.length + "<br>");               // 输出: 3
07. document.write(cars.name + "<br>");                 // 输出: JavaScript
```

Array 对象中的方法

下表中列举了 Array 对象中提供的方法及其描述信息：

方法	描述
concat()	拼接两个或更多的数组，并返回结果
copyWithin()	从数组的指定位置拷贝元素到数组的另一个指定位置中
entries()	返回数组的可迭代对象
every()	检测数值元素的每个元素是否都符合条件
fill()	使用一个固定值来填充数组
filter()	检测数值元素，并返回符合条件所有元素的数组
find()	返回符合传入函数条件的数组元素
findIndex()	返回符合传入函数条件的数组元素索引
forEach()	数组每个元素都执行一次回调函数
from()	通过给定的对象中创建一个数组
includes()	判断一个数组是否包含一个指定的值
indexOf()	搜索数组中的元素，并返回它所在的位置
isArray()	判断对象是否为数组
join()	把数组的所有元素放入一个字符串
keys()	返回数组的可迭代对象，包含原始数组的键（key）
lastIndexOf()	搜索数组中的元素，并返回它最后出现的位置

map()	通过指定函数处理数组的每个元素，并返回处理后的数组
pop()	删除数组的最后一个元素并返回删除的元素
push()	向数组的末尾添加一个或更多元素，并返回数组的长度
reduce()	累加（从左到右）数组中的所有元素，并返回结果
reduceRight()	累加（从右到左）数组中的所有元素，并返回结果
reverse()	反转数组中元素的顺序
shift()	删除并返回数组的第一个元素
slice()	截取数组的一部分，并返回这个新的数组
some()	检测数组元素中是否有元素符合指定条件
sort()	对数组的元素进行排序
splice()	从数组中添加或删除元素
toString()	把数组转换为字符串，并返回结果
unshift()	向数组的开头添加一个或多个元素，并返回新数组的长度
valueOf()	返回数组对象的原始值

示例代码如下：

```
01. var fruits = ["Orange", "Banana", "Apple", "Papaya", "Mango"];
02.
03. document.write(fruits.entries() + "<br>"); // 返回: [object
Array Iterator]
04. document.write(fruits.includes("Apple") + "<br>"); // 返回: true
05. document.write(fruits.fill("grape") + "<br>"); // 返回:
grape,grape,grape,grape,grape
06.
07. var fruits = ["Orange", "Banana", "Apple", "Papaya", "Mango"];
08. document.write(fruits.indexOf("Mango") + "<br>"); // 返回: 4
09. document.write(Array.isArray(fruits) + "<br>"); // 返回:
true
10. document.write(fruits.join("-") + "<br>"); // 返回:
Orange-Banana-Apple-Papaya-Mango
11. document.write(fruits.lastIndexOf("Banana") + "<br>"); // 返回: 1
12. document.write(fruits.pop() + "<br>"); // 返回:
Mango
13. document.write(fruits.push("Watermelon") + "<br>"); // 返回: 5
14. document.write(fruits.unshift("Lemon","Pineapple") + "<br>"); // 返回: 7
```

```
15. document.write(fruits.slice(1, 5) + "<br>");           // 返回:
    Pineapple,Orange,Banana,Apple
16. document.write(fruits.sort() + "<br>");                 // 返回:
    Apple,Banana,Lemon,Orange,Papaya,Pineapple,Watermelon
17. document.write(fruits.valueOf() + "<br>");             // 返回:
    Apple,Banana,Lemon,Orange,Papaya,Pineapple,Watermelon
```